```
EEEEEEEEEEEEEEE  DDDDDDDDDDD     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    TTTTTTTTTTTTTTT
EEE              DDD         DDD      TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEEEEEEEEEEE     DDD          DDD     TTT
EEEEEEEEEEEE     DDD          DDD     TTT
EEEEEEEEEEEE     DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDD         TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD        TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD        TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD        TTT
```

FILEID**FILEIO

```
FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    IIIIII      000000
FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    IIIIII      000000
FF              II      LL          EE              II      00      00
FF              II      LL          EE              II      00      00
FF              II      LL          EE              II      00      00
FF              II      LL          EE              II      00      00
FFFFFFFF        II      LL          EEEEEEEE        II      00      00
FFFFFFFF        II      LL          EEEEEEEE        II      00      00
FF              II      LL          EE              II      00      00
FF              II      LL          EE              II      00      00
FF              II      LL          EE              II      00      00      ....
FF              II      LL          EE              II      00      00      ....
FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE    IIIIII      000000      ....
FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE    IIIIII      000000      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0 %TITLE 'FILEIO - Central file I/O module'
;     2          0002  0 MODULE EDT$FILEIO (                     ! Central file I/O routine for EDT
;     3          0003  0                 IDENT = 'V04-000'               ! File: FILEIO.BLI Edit: JBS1062
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1 !
;     7          0007  1 !**********************************************************************************
;     8          0008  1 !*                                                                                *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                       *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                         *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                           *
;    12          0012  1 !*                                                                                *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED         *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE        *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER        *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY        *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY        *
;    18          0018  1 !*  TRANSFERRED.                                                                   *
;    19          0019  1 !*                                                                                *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE         *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT         *
;    22          0022  1 !*  CORPORATION.                                                                   *
;    23          0023  1 !*                                                                                *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS        *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                        *
;    26          0026  1 !*                                                                                *
;    27          0027  1 !*                                                                                *
;    28          0028  1 !**********************************************************************************
;    29          0029  1 !
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY:     EDT -- The DEC Standard Editor
;    33          0033  1 !
;    34          0034  1 ! ABSTRACT:
;    35          0035  1 !
;    36          0036  1 !       This is the central file i/o routine used by EDT.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  Runs in user mode on VAX/VMS and non-privileged PDP-11
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: Shelly T. Solomon, CREATION DATE: 07-Dec-1981
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original.  STS 25-Dec-1981
;    45          0045  1 ! 1-002 - Change module name to EDT$FILEIO.  STS 25-Dec-1981
;    46          0046  1 ! 1-003 - Add calls for include file.  STS 26-Dec-1981
;    47          0047  1 ! 1-004 - Add require files for 11 translations. STS 28-Dec-1981
;    48          0048  1 ! 1-005 - Add linkage attribute to routine.  STS 30-Dec-1981
;    49          0049  1 ! 1-006 - Signal any errors.  STS 06-Jan-1982
;    50          0050  1 ! 1-007 - Add code for opening output file. STS 13-Jan-1982
;    51          0051  1 ! 1-008 - Fix DSC$A_POINTER macro STS 14-Jan-1982
;    52          0052  1 ! 1-009 - Add gets and puts STS 15-Jan-1982
;    53          0053  1 ! 1-010 - Change opening journal file to open in-out. STS 18-Jan-1982
;    54          0054  1 ! 1-011 - Fixed undefined symbol EDT$$opn_inout on 11. STS 19-Jan-1982
;    55          0055  1 ! 1-012 - output filenames with error messages. STS 19-Jan-1982
;    56          0056  1 ! 1-013 - Change the defaulting of the journal file name. STS 21-Jan-1982
;    57          0057  1 ! 1-014 - Add check to see if file is VFC format. STS 22-Jan-1982
```

```
;    58          0058  1 ! 1-015 - fix journal file name for 11's. STS 26-Jan-1982
;    59          0059  1 ! 1-016 - Add dot to sequence parameter passed with journal file.
;    60          0060  1 !           STS 28-Jan-1982
;    61          0061  1 ! 1-017 - Pass RHB info down to 11 i/o routines. STS 02-Feb-1982
;    62          0062  1 ! 1-018 - Take out extra dot in get on 11's, also
;    63          0063  1 !         fix include rab. STS 08-Feb-1982
;    64          0064  1 ! 1-019 - add flush for journal buffer. STS 11-Feb-1982
;    65          0065  1 ! 1-020 - Take out call to edt$$get_fnam. STS 12-Feb-1982
;    66          0066  1 ! 1-021 - Pass correct status back to caller. STS 26-Feb-1982
;    67          0067  1 ! 1-022 - Add literals for callable parameters. STS 08-Mar-1982
;    68          0068  1 ! 1-023 - Fix status passed on opening write file. STS 10-Mar-1982
;    69          0069  1 ! 1-024 - Rearrange interface to EDT$IOMOD to improve the rationality
;    70          0070  1 !         of file naming.  JBS 25-Mar-1982
;    71          0071  1 ! 1-025 - Worry about non-standard input files.  JBS 26-Mar-1982
;    72          0072  1 ! 1-026 - Correct a typo.  JBS 27-Mar-1982
;    73          0073  1 ! 1-027 - Make the new file handling logic work on the PDP-11.  JBS 29-Mar-1982
;    74          0074  1 ! 1-028 - Use temporary file for WRITE and EXIT and then Rename it.  SMB 31-Mar-1982
;    75          0075  1 ! 1-029 - Add related file names for the PDP-11.  JBS 31-Mar-1982
;    76          0076  1 ! 1-030 - Distinguish two cases of output open for journal files on the PDP-11
;    77          0077  1 !         and add a flush counter to improve PDP-11 performance.  JBS 01-Apr-1982
;    78          0078  1 ! 1-031 - Rearrange file name handling for the journal file.  JBS 02-Apr-1982
;    79          0079  1 ! 1-032 - Make more modifications for WRITE/EXIT to temp files.  SMB 02-Apr-1982
;    80          0080  1 ! 1-033 - Cannot use %REF in STRING_DESC.  JBS 03-Apr-1982
;    81          0081  1 ! 1-034 - Fix bugs in PDP-11 opening of output files.  SMB 06-Apr-1982
;    82          0082  1 ! 1-035 - Add rename for PDP-11's and CLOSE_DEL for output files.  SMB 08-Apr-1982
;    83          0083  1 ! 1-036 - Convert PDP-11 filenames to uppercase.  SMB 12-Apr-1982
;    84          0084  1 ! 1-037 - Take out fix 1-036(move to LWRITE)-fix error message filename for VAX.  SMB 13-Apr-1982
;    85          0085  1 ! 1-038 - Always return status when closing PDP-11 files.  JBS 09-Apr-1982
;    86          0086  1 ! 1-039 - Reverse the attributes flag.  JBS 12-Apr-1982
;    87          0087  1 ! 1-040 - Merge the last four edits, which were done independently.  JBS 15-Apr-1982
;    88          0088  1 ! 1-041 - Add a parse before opening output files.  SMB 15-Apr-1982
;    89          0089  1 ! 1-042 - Put back line accidently deleted for filename storage.  SMB 16-Apr-1982
;    90          0090  1 ! 1-043 - Conditionalize the conversion to uppercase.  SMB 22-Apr-1982
;    91          0091  1 ! 1-044 - Restrict renaming to disks or DECtapes only.  SMB  26-Apr-1982
;    92          0092  1 ! 1-045 - Change the ordinals of global literals for file types.  SMB 19-May-1982
;    93          0093  1 ! 1-046 - Add some comments. STS 19-May-1982
;    94          0094  1 ! 1-047 - Clean up the magic numbers.  JBS 25-May-1982
;    95          0095  1 ! 1-048 - Don't use special linkage on 11's. STS 03-Jun-1982
;    96          0096  1 ! 1-049 - On OPEN, use RHB as the default name.  Also, don't use special linkages on
;    97          0097  1 !         VAX either, since the special linkage used by CALLFIO is compatible with
;    98          0098  1 !         the standard VAX/VMS linkage conventions.  JBS 15-Jun-1982
;    99          0099  1 ! 1-050 - Implement the new file defaulting rules.  JBS 17-Jun-1982
;   100          0100  1 ! 1-051 - Signal any bad status from flushing the journal file. STS 30-Jun-1982
;   101          0101  1 ! 1-052 - Fix bad parameter pass in open for output without related names.  SMB 06-Jul-1982
;   102          0102  1 ! 1-053 - Add a special check for RSTS disk files.  SMB 07-Jul-1982
;   103          0103  1 ! 1-054 - Store status on PDP-11 open for output.  SMB 19-Jul-1982
;   104          0104  1 ! 1-055 - Check for errors when deleting the journal file.  JBS 22-Feb-1983
;   105          0105  1 ! 1-056 - Don't maximize version number on WRITE.  JBS 04-Apr-1983
;   106          0106  1 ! 1-057 - Fix a typo in PDP-11 output file opening.  JBS 06-Apr-1983
;   107          0107  1 ! 1-058 - Fix the message given when the journal file fails to open for output.  JBS 02-May-1983
;   108          0108  1 ! 1-059 - Improve the appearance of the listing.  JBS 14-Jun-1983
;   109          0109  1 ! 1-060 - On VMS, if the EXIT file name is empty,
;   110          0110  1 !         use the resultant name from opening the input file.  JBS 29-Jul-1983
;   111          0111  1 ! 1-061 - Fix bug in edit 060--the input name was being discarded too soon if the
;   112          0112  1 !         output open happened after the input file was closed.  JBS 31-Aug-1983
;   113          0113  1 ! 1-062 - Complete edit 061 by storing the input file name even if the file
;   114          0114  1 !         does not open.  JBS 06-Sep-1983
```

```
;  115          0115  1 !--
;  116          0116  1
```

```
;  118          0117  1 %SBTTL 'Declarations'
;  119          0118  1 !
;  120          0119  1 ! TABLE OF CONTENTS:
;  121          0120  1 !
;  122          0121  1
;  123          0122  1 REQUIRE 'EDTSRC:TRAROUNAM';
;  124          0561  1
;  125          0562  1 FORWARD ROUTINE
;  126          0563  1     EDT$FILEIO;
;  127          0564  1
;  128          0565  1 !
;  129          0566  1 ! INCLUDE FILES:
;  130          0567  1 !
;  131          0568  1
;  132          0569  1 REQUIRE 'EDTSRC:EDTREQ';
;  133          0704  1
;  134        L 0705  1 %IF %BLISS (BLISS32)
;  135          0706  1 %THEN
;  136          0707  1
;  137          0708  1 REQUIRE 'EDTSRC:SYSSYM';
;  138          0738  1
;  139          0739  1 %FI
;  140          0740  1
;  141          0741  1 !
;  142          0742  1 ! MACROS:
;  143          0743  1 !
;  144          0744  1 !+
;  145          0745  1 ! Macro for the file type used as a constant.  This is defined as a macro
;  146          0746  1 ! so we can use %CHARCOUNT to pass the length of the string.
;  147          0747  1 !-
;  148          0748  1 !<BLF/NOFORMAT>
;  149          0749  1
;  150          0750  1 MACRO
;  151          0751  1     TEMP_TYP = '.TMP' %;                ! File type for temporary output files (before being renamed)
;  152          0752  1
;  153          0753  1 !<BLF/FORMAT>
;  154          0754  1 !
;  155          0755  1 ! EQUATED SYMBOLS:
;  156          0756  1 !
;  157          0757  1
;  158        L 0758  1 %IF %BLISS (BLISS32)
;  159          0759  1 %THEN
;  160          0760  1
;  161          0761  1 LITERAL
;  162          0762  1     EDT$K_FAC_NO = 133;
;  163          0763  1
;  164          0764  1 %FI
;  165          0765  1
;  166          0766  1 !+
;  167          0767  1 ! These codes need to be defined here because they need to be known at compile
;  168          0768  1 ! time in order to be used in case statements
;  169          0769  1 !-
;  170          0770  1
;  171          0771  1 GLOBAL LITERAL
;  172          0772  1     EDT$K_OPEN_INPUT = 1,                   ! code signifying we wish to open a file for input
;  173          0773  1     EDT$K_OPEN_OUTPUT_SEQ = 2,              ! code signifying we wish to open a sequenced file for output
;  174          0774  1     EDT$K_OPEN_OUTPUT_NOSEQ = 3,            ! code meaning we wish to open a non-sequenced file for output
```

```
;  175          0775  1      EDT$K_OPEN_IN_OUT = 4,                  ! we wish to open a file for both input and output
;  176          0776  1      EDT$K_GET = 5,                          ! code signifying we want to get a record from a file
;  177          0777  1      EDT$K_PUT = 6,                          ! code signifying we want to put a record to a file
;  178          0778  1      EDT$K_CLOSE_DEL = 7,                    ! we want to close the file and then delete it
;  179          0779  1      EDT$K_CLOSE = 8,                        ! we want to close the file
;  180          0780  1      EDT$K_COMMAND_FILE = 1,                 ! code for the startup command file
;  181          0781  1      EDT$K_INPUT_FILE = 2,                   ! code for the main input file
;  182          0782  1      EDT$K_INCLUDE_FILE = 3,                 ! code for an include file
;  183          0783  1      EDT$K_JOURNAL_FILE = 4,                 ! code for the journal file
;  184          0784  1      EDT$K_OUTPUT_FILE = 5,                  ! code for the  output file
;  185          0785  1      EDT$K_WRITE_FILE = 6;                   ! code for an output file being written
;  186          0786  1
;  187          0787  1  LITERAL
;  188          0788  1      FLUSH_LIMIT = 5;                        ! Flush the journal file buffer after this many records
;  189          0789  1
;  190          0790  1  !+
;  191          0791  1  ! The following symbols are for the interface to EDT$$OPN_OFIDEF.  Note that these values
;  192          0792  1  ! are hard-coded into the MACRO-11 modules, and into EDT$IOMOD.
;  193          0793  1  !-
;  194          0794  1
;  195          0795  1  LITERAL
;  196          0796  1      DISK_FILE_NO = 0,                       ! Not a disk file
;  197          0797  1      DISK_FILE_YES = 1,                      ! Is a disk file
;  198          0798  1      DISK_FILE_RSTS = 2,                     ! Is a disk file on RSTS
;  199          0799  1      SEQ_NO = 0,                             ! The file is not to be sequenced
;  200          0800  1      SEQ_YES = 1,                            ! The file is to be sequenced
;  201          0801  1      RELAT_NONE = 0,                         ! There is no related file name
;  202          0802  1      RELAT_INPUT = 1,                        ! The primary input file is used as the related file
;  203          0803  1      ATTR_INPUT = 0,                         ! Take file attributes from the primary input file
;  204          0804  1      ATTR_DEFAULT = 1,                       ! Use EDT's default file attributes
;  205          0805  1      ATTR_JOURNAL = 2;                       ! Use journal file attributes
;  206          0806  1
;  207          0807  1  !
;  208          0808  1  ! OWN STORAGE:
;  209          0809  1  !
;  210          0810  1  !  in the routine
;  211          0811  1  !
;  212          0812  1  ! EXTERNAL REFERENCES:
;  213          0813  1  !
;  214          0814  1  ! in the routine
;  215          0815  1  !
```

```
;  217          0816  1 %SBTTL 'EDT$FILEIO - Central EDT file I/O routine'
;  218          0817  1
;  219          0818  1 GLOBAL ROUTINE EDT$FILEIO (                          ! Central EDT file I/O routine
;  220          0819  1     FILECODE,                                         ! Function code
;  221          0820  1     FILESTRM,                                         ! Channel number
;  222          0821  1     FILE_REC,                                         ! File name, or record descriptor
;  223          0822  1     FILE_RHB                                          ! Default file name, or record header descriptor
;  224          0823  1     ) =
;  225          0824  1
;  226          0825  1 !++
;  227          0826  1 ! FUNCTIONAL DESCRIPTION:
;  228          0827  1 !
;  229          0828  1 ! This is the basic file I/O routine for EDT.  Callable EDT calls this
;  230          0829  1 ! routine to do any I/O if this is the routine passed to it by the calling
;  231          0830  1 ! program.  This is the routine passed to callable EDT by the "real" EDT.
;  232          0831  1 !
;  233          0832  1 ! FORMAL PARAMETERS:
;  234          0833  1 !
;  235          0834  1 ! filecode = address of fullword containing function code defining type of I/O
;  236          0835  1 !   operation to be performed
;  237          0836  1 ! filestream = address of fullword containing stream identifier
;  238          0837  1 ! file_rec = address of string descriptor, i.e. the file name or place to store
;  239          0838  1 !   record read or place to fetch record to be written
;  240          0839  1 ! file_rhb = address of string descriptor for any record prefixes
;  241          0840  1 !
;  242          0841  1 !
;  243          0842  1 ! Note: the default name is not implemented for WRITE/EXIT/PRINT files
;  244          0843  1 !       (because of .TMP logic).  Fortunately, EDT does not pass a default
;  245          0844  1 !       name on these channels.
;  246          0845  1 !
;  247          0846  1 ! IMPLICIT INPUTS:
;  248          0847  1 !
;  249          0848  1 !       EDT$$Z_SYS_PRIRAB
;  250          0849  1 !       EDT$$Z_SYS_JOURAB
;  251          0850  1 !       EDT$$Z_SYS_CMDRAB
;  252          0851  1 !       EDT$$Z_SYS_ALTRAB
;  253          0852  1 !
;  254          0853  1 !
;  255          0854  1 ! IMPLICIT OUTPUTS:
;  256          0855  1 !
;  257          0856  1 !       EDT$$Z_SYS_PRIRAB
;  258          0857  1 !       EDT$$Z_SYS_JOURAB
;  259          0858  1 !       EDT$$Z_SYS_CMDRAB
;  260          0859  1 !       EDT$$Z_SYS_ALTRAB
;  261          0860  1 !
;  262          0861  1 ! COMPLETION STATUS:
;  263          0862  1 !
;  264          0863  1 !       The only error returned, rather than signaled, is EOF.
;  265          0864  1 !
;  266          0865  1 ! SIDE EFFECTS:
;  267          0866  1 !
;  268          0867  1 !       NONE
;  269          0868  1 !
;  270          0869  1 !--
;  271          0870  1
;  272          0871  2     BEGIN
;  273          0872  2
```

```
;  274          0873  2      MAP
;  275          0874  2          FILE_REC : REF BLOCK [, BYTE],
;  276          0875  2          FILE_RHB : REF BLOCK [, BYTE];
;  277          0876  2
;  278          0877  2      EXTERNAL ROUTINE
;  279          0878  2          EDT$$PAR_FNAME,                     ! parse a file name
;  280          0879  2          EDT$$CNV_UPC,                       ! convert to uppercase
;  281          0880  2          EDT$$REN_FI,                        ! renames a file
;  282          0881  2          EDT$$FLUSH_OBUF,                    ! empties journal buffer
;  283          0882  2          EDT$$OPN_IFIDEF,                    ! I/O input open file routine
;  284          0883  2          EDT$$OPN_OFIDEF,                    ! I/O open output file routine
;  285          0884  2          EDT$$CLS_FI,                        ! I/O close file routine
;  286          0885  2          EDT$$RD_IFI,                        ! read a record from a file stream
;  287          0886  2          EDT$$WR_OFI;                        ! Write a record to a file stream
;  288          0887  2
;  289        L 0888  2  %IF %BLISS (BLISS32)
;  290          0889  2  %THEN
;  291          0890  2
;  292          0891  2      EXTERNAL ROUTINE
;  293          0892  2          STR$FREE1_DX,
;  294          0893  2          EDT$$OPN_INOUT,
;  295          0894  2          STR$COPY_DX,
;  296          0895  2          STR$COPY_R;
;  297          0896  2
;  298          0897  2  %FI
;  299          0898  2
;  300          0899  2      EXTERNAL
;  301          0900  2          EDT$$Z_SYS_PRIRAB,
;  302          0901  2          EDT$$Z_SYS_JOURAB,
;  303          0902  2          EDT$$Z_SYS_CMDRAB,
;  304          0903  2          EDT$$Z_SYS_ALTRAB;
;  305          0904  2
;  306          0905  2      MESSAGES ((INPFILOPN, FILNAM, INTERERR, COMFILNEX, COMFILNOP, NOJNLFIL, INPFILNEX, OUTFILCRE, NONSTDFIL)
;  307          0906  2
;  308        L 0907  2  %IF %BLISS (BLISS32)
;  309          0908  2  %THEN
;  310          0909  2  !+
;  311          0910  2  ! Keep the filename descriptor for each file - on VMS it's a dynamic descriptor
;  312          0911  2  !-
;  313          0912  2
;  314          0913  2      OWN
;  315          0914  2          CMD_DESC : BLOCK [8, BYTE]              ! command file
;  316          0915  2              PRESET ( [DSC$B_DTYPE] = DSC$K_DTYPE_T,
;  317          0916  2                       [DSC$B_CLASS] = DSC$K_CLASS_D,
;  318          0917  2                       [DSC$A_POINTER] = 0,
;  319          0918  2                       [DSC$W_LENGTH] = 0),
;  320          0919  2          JOU_DESC : BLOCK [8, BYTE]              ! journal file
;  321          0920  2              PRESET ( [DSC$B_DTYPE] = DSC$K_DTYPE_T,
;  322          0921  2                       [DSC$B_CLASS] = DSC$K_CLASS_D,
;  323          0922  2                       [DSC$A_POINTER] = 0,
;  324          0923  2                       [DSC$W_LENGTH] = 0),
;  325          0924  2          INP_DESC : BLOCK [8, BYTE]              ! primary input file
;  326          0925  2              PRESET ( [DSC$B_DTYPE] = DSC$K_DTYPE_T,
;  327          0926  2                       [DSC$B_CLASS] = DSC$K_CLASS_D,
;  328          0927  2                       [DSC$A_POINTER] = 0,
;  329          0928  2                       [DSC$W_LENGTH] = 0),
;  330          0929  2          ALT_DESC : BLOCK [8, BYTE]              ! temporary or secondary file
```

```
;  331          0930  2              PRESET ( [DSC$B_DTYPE] = DSC$K_DTYPE_T,
;  332          0931  2                       [DSC$B_CLASS] = DSC$K_CLASS_D,
;  333          0932  2                       [DSC$A_POINTER] = 0,
;  334          0933  2                       [DSC$W_LENGTH] = 0),
;  335          0934  2          OUT_DESC : BLOCK [8, BYTE]                 ! output file
;  336          0935  2              PRESET ( [DSC$B_DTYPE] = DSC$K_DTYPE_T,
;  337          0936  2                       [DSC$B_CLASS] = DSC$K_CLASS_D,
;  338          0937  2                       [DSC$A_POINTER] = 0,
;  339          0938  2                       [DSC$W_LENGTH] = 0);
;  340          0939  2
;  341          0940  2      !+
;  342          0941  2      ! The resultant name from the primary input open, used for the primary output open.
;  343          0942  2      ! (We cannot use INP_DESC since it is released after the input file is closed,
;  344          0943  2      ! which may be before the output file is opened.)
;  345          0944  2      !-
;  346          0945  2
;  347          0946  2          OWN
;  348          0947  2              INP_NAME : VECTOR [256, BYTE],
;  349          0948  2              INP_NAME_LEN;
;  350          0949  2
;  351        U 0950  2      %ELSE
;  352        U 0951  2
;  353        U 0952  2          OWN
;  354        U 0953  2              CMD_DESC : BLOCK [8, BYTE]             ! command file
;  355        U 0954  2                  PRESET ( [DSC$A_POINTER] = 0,
;  356        U 0955  2                           [DSC$W_LENGTH] = 0),
;  357        U 0956  2              JOU_DESC : BLOCK [8, BYTE]             ! journal file
;  358        U 0957  2                  PRESET ( [DSC$A_POINTER] = 0,
;  359        U 0958  2                           [DSC$W_LENGTH] = 0),
;  360        U 0959  2              INP_DESC : BLOCK [8, BYTE]             ! main input file
;  361        U 0960  2                  PRESET ( [DSC$A_POINTER] = 0,
;  362        U 0961  2                           [DSC$W_LENGTH] = 0),
;  363        U 0962  2              ALT_DESC : BLOCK [8, BYTE]             ! temporary or secondary file
;  364        U 0963  2                  PRESET ( [DSC$A_POINTER] = 0,
;  365        U 0964  2                           [DSC$W_LENGTH] = 0),
;  366        U 0965  2              OUT_DESC : BLOCK [8, BYTE]             ! output file
;  367        U 0966  2                  PRESET ( [DSC$A_POINTER] = 0,
;  368        U 0967  2                           [DSC$W_LENGTH] = 0);
;  369        U 0968  2
;  370          0969  2      %FI
;  371          0970  2
;  372          0971  2          OWN
;  373          0972  2
;  374        L 0973  2      %IF %BLISS (BLISS32)
;  375          0974  2      %THEN
;  376          0975  2              OUT_IFI,                               ! internal file id for primary output file
;  377          0976  2              JOU_IFI,                               ! internal file id for journal file
;  378          0977  2              INC_IFI,                               ! internal file id for include file
;  379          0978  2              INP_IFI,                               ! internal file id for primary input
;  380          0979  2              CMD_IFI,                               ! internal file id for command file
;  381          0980  2      %FI
;  382          0981  2
;  383          0982  2              DISK_FI,                               ! flag indicating opening a renameable file for output
;  384          0983  2              FLUSH_COUNTER : INITIAL (0),           ! counts PUTs to journal towards flushing the buffer
;  385          0984  2              INCL_VFC,                              ! flag indicating include file is VFC format file
;  386          0985  2              INPUT_VFC;                             ! flag indicating primary input is VFC format file
;  387          0986  2
```

```
;   388          0987  2      LOCAL
;   389          0988  2          VFC,
;   390          0989  2          ERROR,
;   391          0990  2          IO_STS,
;   392          0991  2          IO_STV,
;   393          0992  2          STATUS;
;   394          0993  2
;   395          0994  2      BIND
;   396          0995  2          FILE_DESC = .FILE_REC : BLOCK [, BYTE], ! passed in descriptor for filename or record in or out
;   397          0996  2          RHB_DESC = .FILE_RHB : BLOCK [, BYTE];  ! record header block descriptor
;   398          0997  2
;   399          0998  2  !+
;   400          0999  2  ! Find out first what kind of operation is requested
;   401          1000  2  !-
;   402          1001  2
;   403          1002  2      CASE ..FILECODE FROM EDT$K_OPEN_INPUT TO EDT$K_CLOSE OF
;   404          1003  2          SET
;   405          1004  2  !+
;   406          1005  2  ! Open a file for input
;   407          1006  2  !-
;   408          1007  2
;   409          1008  2          [EDT$K_OPEN_INPUT] :                    ! we want to open a file
;   410          1009  3              BEGIN
;   411          1010  3
;   412          1011  3              LOCAL
;   413          1012  3                  NONSTD;
;   414          1013  3
;   415        L 1014  3  %IF %BLISS (BLISS16)
;   416        U 1015  3  %THEN
;   417        U 1016  3              EDT$$CNV_UPC (.FILE_DESC [DSC$A_POINTER], .FILE_DESC [DSC$W_LENGTH]);
;   418          1017  3  %FI
;   419          1018  3
;   420          1019  3              NONSTD = 0;
;   421          1020  3
;   422          1021  3              CASE ..FILESTRM FROM EDT$K_COMMAND_FILE TO EDT$K_INCLUDE_FILE OF
;   423          1022  3                  SET                             ! which file?
;   424          1023  3
;   425          1024  3                  [EDT$K_COMMAND_FILE] :          ! open the command file for input
;   426          1025  4                      BEGIN
;   427          1026  4
;   428        L 1027  4  %IF %BLISS (BLISS32)
;   429          1028  4  %THEN
;   430          1029  4                      CMD_IFI = EDT$$OPN_IFIDEF (EDT$$Z_SYS_CMDRAB, FILE_DESC, .RHB_DESC [DSC$A_POINTER],
;   431          1030  4                          .RHB_DESC [DSC$W_LENGTH], RELAT_NONE, IO_STS, IO_STV, VFC, NONSTD);
;   432          1031  4  !+
;   433          1032  4  ! If the open failed then find out why
;   434          1033  4  !-
;   435          1034  4
;   436          1035  5                      IF (.CMD_IFI EQL 0)
;   437          1036  4                      THEN
;   438          1037  4  !+
;   439          1038  4  ! Signal an error
;   440          1039  4  !-
;   441          1040  4                          SIGNAL_STOP (SHR$_OPENIN + (EDT$K_FAC_NO*65536) + STS$K_SEVERE, 1, FILE_DESC,
;   442          1041  4                              .IO_STS, .IO_STV);
;   443          1042  4
;   444          1043  4  !+
```

```
;  445          1044  4 ! If the file is non-standard, indicate this.
;  446          1045  4 !-
;  447          1046  4
;  448          1047  4                     IF .NONSTD THEN IO_STS = EDT$_NONSTDFIL;
;  449          1048  4
;  450          1049  4 !+
;  451          1050  4 ! Save the complete filename
;  452          1051  4 !-
;  453          1052  4                     STRING_DESC (CMD_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
;  454        U 1053  4 %ELSE
;  455        U 1054  4                     IO_STS = EDT$$OPN_IFIDEF (EDT$$Z_SYS_CMDRAB, .FILE_DESC [DSC$A_POINTER],
;  456        U 1055  4                         .FILE_DESC [DSC$W_LENGTH], .RHB_DESC [DSC$A_POINTER], .RHB_DESC [DSC$W_LENGTH], 0, 0
;  457        U 1056  4                         0, 0);
;  458          1057  4 %FI
;  459          1058  4
;  460          1059  4                     RETURN (.IO_STS);           ! return status
;  461          1060  3                     END;
;  462          1061  3
;  463          1062  3                 [EDT$K_INPUT_FILE] :            ! open the primary input file for input
;  464          1063  4                     BEGIN
;  465          1064  4
;  466        L 1065  4 %IF %BLISS (BLISS32)
;  467          1066  4 %THEN
;  468          1067  4                     INP_IFI = EDT$$OPN_IFIDEF (EDT$$Z_SYS_PRIRAB, FILE_DESC, .RHB_DESC [DSC$A_POINTER],
;  469          1068  4                         .RHB_DESC [DSC$W_LENGTH], RELAT_NONE, IO_STS, IO_STV, INPUT_VFC, NONSTD);
;  470          1069  4 !+
;  471          1070  4 ! Save the name for opening the output file on VMS, even if the input file does not open.
;  472          1071  4 !-
;  473          1072  4                     INP_NAME_LEN = .FILE_DESC [DSC$W_LENGTH];
;  474          1073  4                     CH$MOVE (.INP_NAME_LEN, .FILE_DESC [DSC$A_POINTER], INP_NAME);
;  475          1074  4 !+
;  476          1075  4 ! Check for open failure.
;  477          1076  4 !-
;  478          1077  4
;  479          1078  5                     IF (.INP_IFI EQL 0)
;  480          1079  4                     THEN
;  481          1080  4                         SIGNAL_STOP (SHR$_OPENIN + (EDT$K_FAC_NO*65536) + STS$K_SEVERE,
;  482          1081  4                             1, FILE_DESC, .IO_STS, .IO_STV);
;  483          1082  4
;  484          1083  4 !+
;  485          1084  4 ! If the file is non-standard, indicate this.
;  486          1085  4 !-
;  487          1086  4
;  488          1087  4                     IF .NONSTD THEN IO_STS = EDT$_NONSTDFIL;
;  489          1088  4
;  490        U 1089  4 %ELSE
;  491        U 1090  4                     IO_STS = EDT$$OPN_IFIDEF (EDT$$Z_SYS_PRIRAB, .FILE_DESC [DSC$A_POINTER],
;  492        U 1091  4                         .FILE_DESC [DSC$W_LENGTH], .RHB_DESC [DSC$A_POINTER], .RHB_DESC [DSC$W_LENGTH], 0, 0
;  493        U 1092  4                         0, 0);
;  494          1093  4 %FI
;  495          1094  4
;  496          1095  4 !+
;  497          1096  4 ! Save the complete filename.  This is needed on the PDP-11 for opening the journal file.
;  498          1097  4 !-
;  499          1098  4                     STRING_DESC (INP_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
;  500          1099  4                     RETURN (.IO_STS);           ! return status
;  501          1100  3                     END;
```

```
:  502          1101  3
:  503          1102  3                    [EDT$K_INCLUDE_FILE] :          ! open include file for input
:  504          1103  4                        BEGIN
:  505          1104  4
:  506        L 1105  4 %IF %BLISS (BLISS32)
:  507          1106  4 %THEN
:  508          1107  5                        BEGIN
:  509          1108  5                        INCL_IFI = EDT$$OPN_IFIDEF (EDT$$Z_SYS_ALTRAB, FILE_DESC, .RHB_DESC [DSC$A_POINTER],
:  510          1109  5                            .RHB_DESC [DSC$W_LENGTH], RELAT_INPUT, IO_STS, IO_STV, INC[_VFC, NONSTD);
:  511          1110  5
:  512          1111  6                        IF (.INCL_IFI EQL 0)
:  513          1112  5                        THEN
:  514          1113  5 !+
:  515          1114  5 ! Signal the error
:  516          1115  5 !-
:  517          1116  5                            SIGNAL_STOP (SHR$_OPENIN + (EDT$K_FAC_NO*65536) + STS$K_SEVERE, 1, FILE_DESC,
:  518          1117  5                                .IO_STS, .IO_STV);
:  519          1118  5
:  520          1119  5 !+
:  521          1120  5 ! If the file is non-standard, indicate this.
:  522          1121  5 !-
:  523          1122  5
:  524          1123  5                        IF .NONSTD THEN IO_STS = EDT$_NONSTDFIL;
:  525          1124  5
:  526          1125  5 !+
:  527          1126  5 ! Save the complete filename
:  528          1127  5 !-
:  529          1128  5                        STRING_DESC (ALT_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
:  530          1129  4                        END;
:  531        U 1130  4 %ELSE
:  532        U 1131  4                        IO_STS = EDT$$OPN_IFIDEF (EDT$$Z_SYS_ALTRAB, .FILE_DESC [DSC$A_POINTER],
:  533        U 1132  4                            .FILE_DESC [DSC$W_LENGTH], .RHB_DESC [DSC$A_POINTER], .RHB_DESC [DSC$W_LENGTH],
:  534        U 1133  4                            .INP_DESC [DSC$A_POINTER], .INP_DESC [DSC$W_LENGTH], 0, 0);
:  535          1134  4 %FI
:  536          1135  4
:  537          1136  4                        RETURN (.IO_STS);
:  538          1137  3                        END;
:  539          1138  3
:  540          1139  3                    [INRANGE, OUTRANGE] :
:  541          1140  3                        ASSERT (0);
:  542          1141  3                    TES;
:  543          1142  3
:  544          1143  2                END;
:  545          1144  2 !+
:  546          1145  2 ! Open a file for output
:  547          1146  2 !-
:  548          1147  2
:  549          1148  2            [EDT$K_OPEN_OUTPUT_SEQ, EDT$K_OPEN_OUTPUT_NOSEQ] :
:  550          1149  3                BEGIN
:  551          1150  3
:  552          1151  3                LOCAL
:  553          1152  3                    SEQ;
:  554          1153  3
:  555        L 1154  3 %IF %BLISS (BLISS16)
:  556        U 1155  3 %THEN
:  557        U 1156  3                EDT$$CNV_UPC (.FILE_DESC [DSC$A_POINTER], .FILE_DESC [DSC$W_LENGTH]);
:  558          1157  3 %FI
```

```
;  559          1158  3
;  560          1159  4              IF (..FILECODE EQL EDT$K_OPEN_OUTPUT_SEQ)
;  561          1160  3              THEN
;  562          1161  3                  SEQ = SEQ_YES                   ! make it a sequenced VFC file
;  563          1162  3              ELSE
;  564          1163  3                  SEQ = SEQ_NO;                   ! not a sequenced file
;  565          1164  3
;  566          1165  3              CASE ..FILESTRM FROM EDT$K_JOURNAL_FILE TO EDT$K_WRITE_FILE OF
;  567          1166  3                  SET
;  568          1167  3
;  569          1168  3                  [EDT$K_OUTPUT_FILE, EDT$K_WRITE_FILE] :         ! WRITE or OUTPUT file
;  570          1169  4                      BEGIN
;  571          1170  4
;  572          1171  4                      LOCAL
;  573          1172  4                          ATT,    ! 0 = use input file attributes, 1 = use EDT's default file attributes
;  574          1173  4                          RELAT,  ! 0 = no related file, 1 = use input file's name and type before default nam
;  575          1174  4                          FORCE_MAXV;             ! 1 = force maximum version number
;  576          1175  4
;  577          1176  5                      IF (..FILESTRM EQL EDT$K_OUTPUT_FILE)
;  578          1177  4                      THEN
;  579          1178  5                          BEGIN
;  580          1179  5                          ATT = ATTR_INPUT;
;  581          1180  5                          RELAT = RELAT_INPUT;
;  582          1181  5                          FORCE_MAXV = T;
;  583          1182  5                          END
;  584          1183  4                      ELSE
;  585          1184  5                          BEGIN
;  586          1185  5                          ATT = ATTR_DEFAULT;
;  587          1186  5                          RELAT = RELAT_NONE;
;  588          1187  5                          FORCE_MAXV = 0;
;  589          1188  4                          END;
;  590          1189  4
;  591          1190  4 !+
;  592          1191  4 ! This code cannot handle default file names, so make sure there isn't one.
;  593          1192  4 !-
;  594          1193  4                      ASSERT (.RHB_DESC [DSC$W_LENGTH] EQL 0);
;  595          1194  4                      DISK_FI = 0;
;  596          1195  4
;  597        L 1196  4 %IF %BLISS (BLISS32)
;  598          1197  4 %THEN
;  599          1198  4 !+
;  600          1199  4 ! On VMS, if the EXIT file name is not specified, use the resultant file name from the input open.
;  601          1200  4 ! Because we are forcing maximum version number the version number in the input file name string
;  602          1201  4 ! won't cause trouble.
;  603          1202  4 !-
;  604          1203  4
;  605          1204  5                      IF ((.RELAT EQL RELAT_INPUT) AND (.FILE_DESC [DSC$W_LENGTH] EQLU 0))
;  606          1205  4                      THEN
;  607          1206  5                          BEGIN
;  608          1207  5                          STRING_DESC (FILE_DESC, INP_NAME_LEN, INP_NAME);
;  609          1208  4                          END;
;  610          1209  4
;  611          1210  4 !+
;  612          1211  4 ! Parse the output file name - If successful, then do the open; otherwise
;  613          1212  4 ! signal an error on open
;  614          1213  4 !-
;  615          1214  4                      STATUS = EDT$$PAR_FNAME (EDT$$Z_SYS_ALTRAB, FILE_DESC, .RELAT, DISK_FI, IO_STS, IO_STV);
```

```
;  616          1215  4
;  617          1216  5                      IF ( NOT .STATUS)
;  618          1217  4                      THEN
;  619          1218  4                          SIGNAL_STOP (SHR$_OPENOUT + (EDT$K_FAC_NO*65536) + STS$K_SEVERE,
;  620          1219  4                              1, FILE_DESC, .IO_STS, .IO_STV);
;  621          1220  4
;  622          1221  4                      OUT_DESC [DSC$W_LENGTH] = 0;
;  623          1222  4                      OUT_DESC [DSC$A_POINTER] = 0;
;  624          1223  4  !+
;  625          1224  4  ! Save description of output file before translation with .TMP extension
;  626          1225  4  ! if this is a disk or DECtape file for rename later
;  627          1226  4  !-
;  628          1227  4
;  629          1228  5                      IF (.DISK_FI)
;  630          1229  4                      THEN
;  631          1230  5                          BEGIN
;  632          1231  5                          STRING_DESC (OUT_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
;  633          1232  5                          STR$COPY_R (FILE_DESC, %REF (%CHARCOUNT (TEMP_TYP)), UPLIT (BYTE (TEMP_TYP)));
;  634          1233  5                          FORCE_MAXV = 1;          ! For .TMP file, force max version number
;  635          1234  4                          END;
;  636          1235  4
;  637          1236  4  !+
;  638          1237  4  ! If this is a disk file, open a temporary file for output, then rename later
;  639          1238  4  ! if all goes well.  If not a disk file, just open the "given" file.
;  640          1239  4  !-
;  641          1240  4                      OUT_IFI = EDT$$OPN_OFIDEF (EDT$$Z_SYS_ALTRAB, FILE_DESC, .OUT_DESC [DSC$A_POINTER],
;  642          1241  4                          .OUT_DESC [DSC$W_LENGTH], .SEQ, .RELAT, .ATT, .FORCE_MAXV, IO_STS, IO_STV);
;  643          1242  4  !+
;  644          1243  4  ! Signal an error
;  645          1244  4  !-
;  646          1245  4
;  647          1246  5                      IF (.OUT_IFI EQL 0)
;  648          1247  4                      THEN
;  649          1248  4                          SIGNAL_STOP (SHR$_OPENOUT + (EDT$K_FAC_NO*65536) + STS$K_SEVERE,
;  650          1249  4                              1, FILE_DESC, .IO_STS, .IO_STV);
;  651          1250  4
;  652          1251  4  !+
;  653          1252  4  ! Save the complete filename for the close later
;  654          1253  4  !-
;  655          1254  4                      STRING_DESC (ALT_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
;  656        U 1255  4  %ELSE
;  657        U 1256  4
;  658        U 1257  4                      IF (.RELAT EQL RELAT_INPUT)
;  659        U 1258  4                      THEN
;  660        U 1259  4                          BEGIN
;  661        U 1260  4                          STATUS = EDT$$PAR_FNAME (EDT$$Z_SYS_ALTRAB, .FILE_DESC [DSC$A_POINTER],
;  662        U 1261  4                              .FILE_DESC [DSC$W_LENGTH], .INP_DESC [DSC$A_POINTER], .INP_DESC [DSC$W_LENGTH],
;  663        U 1262  4                              DISK_FI);
;  664        U 1263  4                          END
;  665        U 1264  4                      ELSE
;  666        U 1265  4                          BEGIN
;  667        U 1266  4                          STATUS = EDT$$PAR_FNAME (EDT$$Z_SYS_ALTRAB, .FILE_DESC [DSC$A_POINTER],
;  668        U 1267  4                              .FILE_DESC [DSC$W_LENGTH], 0, 0, DISK_FI);
;  669        U 1268  4                          END;
;  670        U 1269  4
;  671        U 1270  4                      STRING_DESC (OUT_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
;  672        U 1271  4
```

```
;  673        U 1272  4                    IF (.STATUS)
;  674        U 1273  4                    THEN
;  675        U 1274  4 !+
;  676        U 1275  4 ! Disk files are handled specially on RSTS.  We don't use a .TMP extension
;  677        U 1276  4 ! but rather open it in temporary mode using the actual name given
;  678        U 1277  4 !-
;  679        U 1278  4
;  680        U 1279  4                        IF (.DISK_FI EQL DISK_FILE_YES)
;  681        U 1280  4                        THEN
;  682        U 1281  4                            BEGIN
;  683        U 1282  4
;  684        U 1283  4                            IF (.RELAT EQL RELAT_INPUT)
;  685        U 1284  4                            THEN
;  686        U 1285  4                                BEGIN
;  687        U 1286  4                                IO_STS = EDT$$OPN_OFIDEF (EDT$$Z_SYS_ALTRAB, UPLIT (BYTE (TEMP_TYP)),
;  688        U 1287  4                                    %CHARCOUNT (TEMP_TYP), .FILE_DESC [DSC$A_POINTER],
;  689        U 1288  4                                    .FILE_DESC [DSC$W_LENGTH], .INP_DESC [DSC$A_POINTER],
;  690        U 1289  4                                    .INP_DESC [DSC$W_LENGTH], 1, 0, .SEQ, .ATT);
;  691        U 1290  4                                END
;  692        U 1291  4                            ELSE
;  693        U 1292  4                                BEGIN
;  694        U 1293  4                                IO_STS = EDT$$OPN_OFIDEF (EDT$$Z_SYS_ALTRAB, UPLIT (BYTE (TEMP_TYP)),
;  695        U 1294  4                                    %CHARCOUNT (TEMP_TYP), .FILE_DESC [DSC$A_POINTER],
;  696        U 1295  4                                    .FILE_DESC [DSC$W_LENGTH], 0, 0, 1, 0, .SEQ, .ATT);
;  697        U 1296  4                                END;
;  698        U 1297  4
;  699        U 1298  4                            END
;  700        U 1299  4                        ELSE
;  701        U 1300  4                            BEGIN
;  702        U 1301  4
;  703        U 1302  4                            IF (.RELAT EQL RELAT_INPUT)
;  704        U 1303  4                            THEN
;  705        U 1304  4                                BEGIN
;  706        U 1305  4                                IO_STS = EDT$$OPN_OFIDEF (EDT$$Z_SYS_ALTRAB, .FILE_DESC [DSC$A_POINTER],
;  707        U 1306  4                                    .FILE_DESC [DSC$W_LENGTH], 0, 0, .INP_DESC [DSC$A_POINTER],
;  708        U 1307  4                                    .INP_DESC [DSC$W_LENGTH], .FORCE_MAXV, 0, .SEQ, .ATT);
;  709        U 1308  4                                END
;  710        U 1309  4                            ELSE
;  711        U 1310  4                                BEGIN
;  712        U 1311  4                                IO_STS = EDT$$OPN_OFIDEF (EDT$$Z_SYS_ALTRAB, .FILE_DESC [DSC$A_POINTER],
;  713        U 1312  4                                    .FILE_DESC [DSC$W_LENGTH], 0, 0, 0, 0, .FORCE_MAXV, 0, .SEQ, .ATT);
;  714        U 1313  4                                END;
;  715        U 1314  4
;  716        U 1315  4                            END
;  717        U 1316  4
;  718        U 1317  4                        ELSE
;  719        U 1318  4                            IO_STS = .STATUS;
;  720        U 1319  4
;  721          1320  4 %FI
;  722          1321  4
;  723          1322  4                        RETURN (.IO_STS);
;  724          1323  3                        END;
;  725          1324  3
;  726          1325  3                    [EDT$K_JOURNAL_FILE] :
;  727          1326  4                        BEGIN
;  728          1327  4
;  729        L 1328  4 %IF %BLISS (BLISS32)
```

```
;  730          1329  4 %THEN
;  731          1330  4                       JOU_IFI = EDT$$OPN_OFIDEF (EDT$$Z_SYS_JOURAB, FILE_DESC, .RHB_DESC [DSC$A_POINTER],
;  732          1331  4                           .RHB_DESC [DSC$W_LENGTH], SEQ_NO, RELAT_INPUT, ATTR_JOURNAL, 1, IO_STS, IO_STV);
;  733          1332  4
;  734          1333  5                       IF (.JOU_IFI EQL 0)
;  735          1334  4                       THEN
;  736          1335  4                           SIGNAL_STOP (SHR$_OPENOUT + (EDT$K_FAC_NO*65536) + STS$K_SEVERE,
;  737          1336  4                               1, FILE_DESC, .IO_STS, .IO_STV);
;  738          1337  4
;  739          1338  4                       STRING_DESC (JOU_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
;  740        U 1339  4 %ELSE
;  741        U 1340  4 !+
;  742        U 1341  4 ! Note that .SEQ+1 is used to specify a normal output open or an open for append.
;  743        U 1342  4 !-
;  744        U 1343  4                       IO_STS = EDT$$OPN_OFIDEF (EDT$$Z_SYS_JOURAB, .FILE_DESC [DSC$A_POINTER],
;  745        U 1344  4                           .FILE_DESC [DSC$W_LENGTH], .RHB_DESC [DSC$A_POINTER], .RHB_DESC [DSC$W_LENGTH],
;  746        U 1345  4                           .INP_DESC [DSC$A_POINTER], .INP_DESC [DSC$W_LENGTH], 1, .SEQ + 1, 0, 1);
;  747          1346  4 %FI
;  748          1347  4
;  749          1348  4                       RETURN (.IO_STS);
;  750          1349  3                       END;
;  751          1350  3
;  752          1351  3                   [INRANGE, OUTRANGE] :
;  753          1352  3                       ASSERT (0);
;  754          1353  3                   TES;
;  755          1354  3
;  756          1355  2               END;
;  757          1356  2 !+
;  758          1357  2 ! Open a file for both input and output
;  759          1358  2 !-
;  760          1359  2
;  761          1360  2           [EDT$K_OPEN_IN_OUT] :
;  762          1361  3               BEGIN
;  763          1362  3 !+
;  764          1363  3 ! The journal file is the only file we can open this way
;  765          1364  3 !-
;  766          1365  3
;  767          1366  4               IF (..FILESTRM EQL EDT$K_JOURNAL_FILE)
;  768          1367  3               THEN
;  769          1368  4                   BEGIN
;  770          1369  4
;  771        L 1370  4 %IF %BLISS (BLISS32)
;  772          1371  4 %THEN
;  773          1372  4                       JOU_IFI = EDT$$OPN_INOUT (EDT$$Z_SYS_JOURAB, FILE_DESC, .RHB_DESC [DSC$A_POINTER],
;  774          1373  4                           .RHB_DESC [DSC$W_LENGTH], IO_STS, IO_STV);
;  775          1374  4
;  776          1375  5                       IF (.JOU_IFI EQL 0)
;  777          1376  4                       THEN
;  778          1377  4                           SIGNAL_STOP (SHR$_OPENIN + (EDT$K_FAC_NO*65536) + STS$K_SEVERE, 1,
;  779          1378  4                               FILE_DESC, .IO_STS, .IO_STV);
;  780          1379  4
;  781          1380  4                       STRING_DESC (JOU_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
;  782        U 1381  4 %ELSE
;  783        U 1382  4                       IO_STS = EDT$$OPN_IFIDEF (EDT$$Z_SYS_JOURAB, .FILE_DESC [DSC$A_POINTER],
;  784        U 1383  4                           .FILE_DESC [DSC$W_LENGTH], .RHB_DESC [DSC$A_POINTER], .RHB_DESC [DSC$W_LENGTH],
;  785        U 1384  4                           .INP_DESC [DSC$A_POINTER], .INP_DESC [DSC$W_LENGTH], 0, 1);
;  786          1385  4 %FI
```

```
;  787          1386  4
;  788          1387  4                    RETURN (.IO_STS);
;  789          1388  4                    END
;  790          1389                  ELSE
;  791          1390                      ASSERT (0);
;  792          1391
;  793          1392                  END;
;  794          1393
;  795          1394              [EDT$K_GET] :                       ! We wish to get a record from a file
;  796          1395                  BEGIN
;  797          1396
;  798          1397                  LOCAL
;  799          1398                      DESC_ADDR,
;  800          1399                      RAB;
;  801          1400
;  802          1401                  CASE ..FILESTRM FROM EDT$K_COMMAND_FILE TO EDT$K_JOURNAL_FILE OF
;  803          1402                      SET
;  804          1403
;  805          1404                      [EDT$K_COMMAND_FILE] :          ! the startup command file
;  806          1405  4                       BEGIN
;  807          1406  4                       DESC_ADDR = CMD_DESC;
;  808          1407  4                       RAB = EDT$$Z_SYS_CMDRAB;
;  809          1408  4                       VFC = 0;
;  810          1409  4                       END;
;  811          1410
;  812          1411  4                   [EDT$K_INPUT_FILE] :            ! get a record from the primary input file
;  813          1412  4                       BEGIN
;  814          1413  4                       DESC_ADDR = INP_DESC;
;  815          1414  4                       VFC = .INPUT_VFC;
;  816          1415  4                       RAB = EDT$$Z_SYS_PRIRAB;
;  817          1416  4                       END;
;  818          1417
;  819          1418  3                   [EDT$K_INCLUDE_FILE] :          ! the secondary input file
;  820          1419  4                       BEGIN
;  821          1420  4                       VFC = .INCL_VFC;
;  822          1421  4                       DESC_ADDR = ALT_DESC;
;  823          1422  4                       RAB = EDT$$Z_SYS_ALTRAB;
;  824          1423  4                       END;
;  825          1424
;  826          1425  3                   [EDT$K_JOURNAL_FILE] :          ! get a record from the journal file
;  827          1426  4                       BEGIN
;  828          1427  4                       VFC = 0;
;  829          1428  4                       DESC_ADDR = JOU_DESC;
;  830          1429  4                       RAB = EDT$$Z_SYS_JOURAB
;  831          1430  4                       END;
;  832          1431
;  833          1432                      [INRANGE, OUTRANGE] :
;  834          1433                          ASSERT (0);
;  835          1434                      TES;
;  836          1435
;  837        L 1436  %IF %BLISS (BLISS32)
;  838          1437  %THEN
;  839          1438                  STATUS = EDT$$RD_IFI (.RAB, FILE_DESC, RHB_DESC, IO_STS, IO_STV, .VFC);
;  840          1439
;  841          1440  4               IF ( NOT .STATUS)
;  842          1441  3               THEN
;  843          1442  3
```

```
;  844          1443  4                  IF (.IO_STS EQL RMS$_EOF)
;  845          1444  4                  THEN
;  846          1445  4                      RETURN (.IO_STS)
;  847          1446  4                  ELSE
;  848          1447  4                      SIGNAL_STOP (SHR$_READERR + (EDT$K_FAC_NO*65536) + STS$K_SEVERE, 1, .DESC_ADDR, .IO_STS,
;  849          1448  4                          .IO_STV);
;  850          1449  4
;  851        U 1450  4              %ELSE
;  852        U 1451  4                  BEGIN
;  853        U 1452  4
;  854        U 1453  4                  LOCAL
;  855        U 1454  4                      REC_ADDR,
;  856        U 1455  4                      REC_LEN;
;  857        U 1456  4
;  858        U 1457  4                  STATUS = EDT$$RD_IFI (.RAB, REC_ADDR, REC_LEN, .RHB_DESC [DSC$A_POINTER],   !
;  859        U 1458  4                      RHB_DESC [DSC$W_LENGTH]);
;  860        U 1459  4                  STRING_DESC (FILE_DESC, REC_LEN, .REC_ADDR);
;  861        U 1460  4                  END;
;  862          1461  4              %FI
;  863          1462  4
;  864          1463  4                  RETURN (.STATUS);
;  865          1464  4                  END;
;  866          1465  3
;  867          1466  3              [EDT$K_PUT] :                           ! we wish to put a record to a file
;  868          1467  4                  BEGIN
;  869          1468  4
;  870          1469  4                  LOCAL
;  871          1470  4                      DESC_ADDR,
;  872          1471  4                      RAB;
;  873          1472  4
;  874          1473  4                  CASE ..FILESTRM FROM EDT$K_JOURNAL_FILE TO EDT$K_WRITE_FILE OF
;  875          1474  4                      SET
;  876          1475  4
;  877          1476  4                      [EDT$K_OUTPUT_FILE, EDT$K_WRITE_FILE] :         ! put a record in an output file
;  878          1477  5                          BEGIN
;  879          1478  5                          DESC_ADDR = ALT_DESC;
;  880          1479  5                          RAB = EDT$$Z_SYS_ALTRAB;
;  881          1480  5                          END;
;  882          1481  4
;  883          1482  4                      [EDT$K_JOURNAL_FILE] :          ! put a record to the journal file
;  884          1483  5                          BEGIN
;  885          1484  5                          DESC_ADDR = JOU_DESC;
;  886          1485  5                          RAB = EDT$$Z_SYS_JOURAB;
;  887          1486  5                          END;
;  888          1487  4
;  889          1488  4                      [INRANGE, OUTRANGE] :
;  890          1489  4                          ASSERT (0);
;  891          1490  4                      TES;
;  892          1491  4
;  893        L 1492  4              %IF %BLISS (BLISS32)
;  894          1493  4              %THEN
;  895          1494  4                  STATUS = EDT$$WR_OFI (.RAB, FILE_DESC, RHB_DESC, IO_STS, IO_STV);
;  896        U 1495  4              %ELSE
;  897        U 1496  4                  STATUS = EDT$$WR_OFI (.RAB, .FILE_DESC [DSC$A_POINTER], .FILE_DESC [DSC$W_LENGTH],
;  898        U 1497  4                      .RHB_DESC [DSC$A_POINTER]);
;  899          1498  4              %FI
;  900          1499  4
```

```
;  901          1500  4                IF ( NOT .STATUS)
;  902          1501  4                THEN
;  903          1502  4
;  904        L 1503  4 %IF %BLISS (BLISS32)
;  905          1504  4 %THEN
;  906          1505  4                    SIGNAL_STOP (SHR$_WRITEERR + (EDT$K_FAC_NO*65536) + STS$K_SEVERE, 1, .DESC_ADDR, .IO_STS,
;  907          1506  4                        .IO_STV)
;  908          1507  4 %FI
;  909          1508  4
;  910          1509  4                ELSE
;  911          1510  4
;  912          1511  4                    IF (..FILESTRM EQL EDT$K_JOURNAL_FILE)
;  913          1512  4                    THEN                            ! keep the journal buffer clear
;  914          1513  4                        BEGIN
;  915          1514  4                        FLUSH_COUNTER = .FLUSH_COUNTER + 1;
;  916          1515  4
;  917          1516  5                        IF (.FLUSH_COUNTER EQL FLUSH_LIMIT)
;  918          1517  4                        THEN
;  919          1518  5                            BEGIN
;  920          1519  5
;  921        L 1520  5 %IF %BLISS (BLISS32)
;  922          1521  5 %THEN
;  923          1522  5                                STATUS = EDT$$FLUSH_OBUF (.RAB, IO_STV);
;  924          1523  5
;  925          1524  6                                IF ( NOT .STATUS)
;  926          1525  5                                THEN
;  927          1526  5                                    SIGNAL_STOP (SHR$_WRITEERR + (EDT$K_FAC_NO*65536) + STS$K_SEVERE, 1, .DESC_ADDR,
;  928          1527  5                                        .STATUS, .IO_STV);
;  929          1528  5
;  930        U 1529  5 %ELSE
;  931        U 1530  5                                STATUS = EDT$$FLUSH_OBUF (.RAB);
;  932          1531  5 %FI
;  933          1532  5
;  934          1533  5                                FLUSH_COUNTER = 0;
;  935          1534  4                                END;
;  936          1535  4
;  937          1536  3                            END;
;  938          1537  3
;  939          1538  2                    RETURN (.STATUS);
;  940          1539  2                    END;
;  941          1540  2
;  942          1541  2                [EDT$K_CLOSE] :                     ! close a file
;  943          1542  2                    BEGIN
;  944          1543  2
;  945          1544  2                    LOCAL
;  946          1545  2                        DESC_ADDR,
;  947          1546  2                        ERROR;
;  948          1547  2
;  949          1548  2                    CASE ..FILESTRM FROM EDT$K_COMMAND_FILE TO EDT$K_WRITE_FILE OF
;  950          1549  2                        SET
;  951          1550  2
;  952          1551  3                        [EDT$K_COMMAND_FILE] :      ! close the command file
;  953          1552  4                            BEGIN
;  954          1553  4
;  955        L 1554  4 %IF %BLISS (BLISS32)
;  956          1555  4 %THEN
;  957          1556  4                            DESC_ADDR = CMD_DESC;
```

```
;  958          1557  4                          ERROR = SHR$_CLOSEIN;
;  959          1558  4                          EDT$$CLS_FI (.CMD_IFI, EDT$$Z_SYS_CMDRAB, 0, .DESC_ADDR, IO_STS, IO_STV);
;  960        U 1559  4 %ELSE
;  961        U 1560  4                          IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_CMDRAB, 0);
;  962          1561  4 %FI
;  963          1562  4
;  964          1563  3                          END;
;  965          1564  3
;  966          1565  3                      [EDT$K_INPUT_FILE] :            ! close the primary input ifle
;  967          1566  4                          BEGIN
;  968          1567  4
;  969        L 1568  4 %IF %BLISS (BLISS32)
;  970          1569  4 %THEN
;  971          1570  4                          DESC_ADDR = INP_DESC;
;  972          1571  4                          ERROR = SHR$_CLOSEIN;
;  973          1572  4                          EDT$$CLS_FI (.INP_IFI, EDT$$Z_SYS_PRIRAB, 0, .DESC_ADDR, IO_STS, IO_STV);
;  974        U 1573  4 %ELSE
;  975        U 1574  4                          IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_PRIRAB, 0);
;  976          1575  4 %FI
;  977          1576  4
;  978          1577  3                          END;
;  979          1578  3
;  980          1579  3                      [EDT$K_INCLUDE_FILE] :          ! close the secondary input file
;  981          1580  4                          BEGIN
;  982          1581  4
;  983        L 1582  4 %IF %BLISS (BLISS32)
;  984          1583  4 %THEN
;  985          1584  4                          DESC_ADDR = ALT_DESC;
;  986          1585  4                          EDT$$CLS_FI (.INCL_IFI, EDT$$Z_SYS_ALTRAB, 0, .DESC_ADDR, IO_STS, IO_STV);
;  987          1586  4                          ERROR = SHR$_CLOSEIN;
;  988        U 1587  4 %ELSE
;  989        U 1588  4                          IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_ALTRAB, 0);
;  990          1589  4 %FI
;  991          1590  4
;  992          1591  3                          END;
;  993          1592  3
;  994          1593  3                      [EDT$K_OUTPUT_FILE, EDT$K_WRITE_FILE] :         ! close an output file
;  995          1594  4                          BEGIN
;  996          1595  4
;  997          1596  4                          LOCAL
;  998          1597  4                              FORCE_MAXV;
;  999          1598  4
; 1000          1599  4                          IF (..FILESTRM EQL EDT$K_OUTPUT_FILE) THEN FORCE_MAXV = 1 ELSE FORCE_MAXV = 0;
; 1001          1600  4
; 1002        L 1601  4 %IF %BLISS (BLISS32)
; 1003          1602  4 %THEN
; 1004          1603  4                          DESC_ADDR = ALT_DESC;
; 1005          1604  4                          ERROR = SHR$_CLOSEOUT;
; 1006          1605  4                          EDT$$CLS_FI (.OUT_IFI, EDT$$Z_SYS_ALTRAB, 0, .DESC_ADDR, IO_STS, IO_STV);
; 1007          1606  4 !+
; 1008          1607  4 ! Check the status from the close
; 1009          1608  4 !-
; 1010          1609  4
; 1011          1610  5                          IF (.IO_STS)
; 1012          1611  4                          THEN
; 1013          1612  4
; 1014          1613  5                              IF (.DISK_FI)
```

```
;  1015          1614  4                                THEN
;  1016          1615  5                                    BEGIN
;  1017          1616  5                                    EDT$$REN_FI (ALT_DESC, OUT_DESC, .FORCE_MAXV, IO_STS, IO_STV);
;  1018          1617  5                                    STRING_DESC (FILE_DESC, OUT_DESC [DSC$W_LENGTH], .OUT_DESC [DSC$A_POINTER]);
;  1019          1618  5                                    END
;  1020          1619  4                                ELSE
;  1021          1620  4                                    STRING_DESC (FILE_DESC, ALT_DESC [DSC$W_LENGTH], .ALT_DESC [DSC$A_POINTER]);
;  1022          1621  4
;  1023        U 1622  4 %ELSE
;  1024        U 1623  4
;  1025        U 1624  4                            IF (.DISK_FI NEQ DISK_FILE_RSTS) THEN IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_ALTRAB, 0);
;  1026        U 1625  4
;  1027        U 1626  4 !+
;  1028        U 1627  4 ! If this is a disk file and we had a successful close, then rename the
;  1029        U 1628  4 ! temp file to the name originally given
;  1030        U 1629  4 !-
;  1031        U 1630  4
;  1032        U 1631  4                            IF ((.IO_STS) AND (.DISK_FI EQL DISK_FILE_YES))
;  1033        U 1632  4                            THEN
;  1034        U 1633  4                                IO_STS = EDT$$REN_FI (EDT$$Z_SYS_ALTRAB, .OUT_DESC [DSC$A_POINTER],
;  1035        U 1634  4                                    .OUT_DESC [DSC$W_LENGTH], .FORCE_MAXV);
;  1036        U 1635  4
;  1037        U 1636  4 !+
;  1038        U 1637  4 ! If this is a RSTS disk file then do a rename of any currently existing
;  1039        U 1638  4 ! files with the originally given name to the same name with a .BAK
;  1040        U 1639  4 ! extension and close the tentative output file making it permanent
;  1041        U 1640  4 !-
;  1042        U 1641  4
;  1043        U 1642  4                            IF (.DISK_FI EQL DISK_FILE_RSTS)
;  1044        U 1643  4                            THEN
;  1045        U 1644  4                                BEGIN
;  1046        U 1645  4                                IO_STS = EDT$$REN_FI (EDT$$Z_SYS_ALTRAB, .OUT_DESC [DSC$A_POINTER],
;  1047        U 1646  4                                    .OUT_DESC [DSC$W_LENGTH], .FORCE_MAXV);
;  1048        U 1647  4
;  1049        U 1648  4                                IF (.IO_STS) THEN IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_ALTRAB, 0);
;  1050        U 1649  4
;  1051        U 1650  4                                END;
;  1052        U 1651  4
;  1053          1652  4 %FI
;  1054          1653  4
;  1055          1654  3                            END;
;  1056          1655  3
;  1057          1656  3                        [EDT$K_JOURNAL_FILE] :          ! close the journal file
;  1058          1657  4                            BEGIN
;  1059          1658  4
;  1060        L 1659  4 %IF %BLISS (BLISS32)
;  1061          1660  4 %THEN
;  1062          1661  4                            DESC_ADDR = JOU_DESC;
;  1063          1662  4                            ERROR = SHR$_CLOSEOUT;
;  1064          1663  4                            EDT$$CLS_FI (.JOU_IFI, EDT$$Z_SYS_JOURAB, 0, .DESC_ADDR, IO_STS, IO_STV);
;  1065        U 1664  4 %ELSE
;  1066        U 1665  4                            IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_JOURAB, 0);
;  1067          1666  4 %FI
;  1068          1667  4
;  1069          1668  3                            END;
;  1070          1669  3
;  1071          1670  3                        [INRANGE, OUTRANGE] :
```

```
;  1072          1671  3                    ASSERT (0);
;  1073          1672  3                TES;
;  1074          1673  3
;  1075        L 1674  3    %IF %BLISS (BLISS32)
;  1076          1675  3    %THEN
;  1077          1676  3    !+
;  1078          1677  3    ! Check the status from either the close or the rename of output files
;  1079          1678  3    !-
;  1080          1679  3
;  1081          1680  3            IF ( NOT .IO_STS)
;  1082          1681  3            THEN
;  1083          1682  3                SIGNAL_STOP (.ERROR + (EDT$K_FAC_NO*65536) + STS$K_SEVERE, 1, .DESC_ADDR,
;  1084          1683  3                    .IO_STS, .IO_STV);
;  1085          1684  3
;  1086          1685  3            STR$FREE1_DX (.DESC_ADDR);
;  1087          1686  3    %FI
;  1088          1687  3
;  1089          1688  3            RETURN (.IO_STS);
;  1090          1689  3            END;
;  1091          1690  3
;  1092          1691  3          [EDT$K_CLOSE_DEL] :
;  1093          1692  3            BEGIN
;  1094          1693  3
;  1095          1694  3            LOCAL
;  1096          1695  3                DESC_ADDR;
;  1097          1696  3
;  1098          1697  3            CASE ..FILESTRM FROM EDT$K_JOURNAL_FILE TO EDT$K_WRITE_FILE OF
;  1099          1698  3                SET
;  1100          1699  3
;  1101          1700  3                [EDT$K_OUTPUT_FILE, EDT$K_WRITE_FILE] :
;  1102          1701  4                    BEGIN
;  1103          1702  4
;  1104        L 1703  4    %IF %BLISS (BLISS32)
;  1105          1704  4    %THEN
;  1106          1705  4                        DESC_ADDR = ALT_DESC;
;  1107          1706  4                        EDT$$CLS_FI (.OUT_IFI, EDT$$Z_SYS_ALTRAB, 1, ALT_DESC, IO_STS, IO_STV);
;  1108        U 1707  4    %ELSE
;  1109        U 1708  4                        IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_ALTRAB, 1);
;  1110          1709  4    %FI
;  1111          1710  4
;  1112          1711  3                    END;
;  1113          1712  3
;  1114          1713  3                [EDT$K_JOURNAL_FILE] :
;  1115          1714  4                    BEGIN
;  1116          1715  4
;  1117        L 1716  4    %IF %BLISS (BLISS32)
;  1118          1717  4    %THEN
;  1119          1718  4                        DESC_ADDR = JOU_DESC;
;  1120          1719  4                        EDT$$CLS_FI (.JOU_IFI, EDT$$Z_SYS_JOURAB, 2, JOU_DESC, IO_STS, IO_STV);
;  1121        U 1720  4    %ELSE
;  1122        U 1721  4                        IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_JOURAB, 2);
;  1123          1722  4    %FI
;  1124          1723  4
;  1125          1724  3                    END;
;  1126          1725  3
;  1127          1726  3                [INRANGE, OUTRANGE] :
;  1128          1727  3                    ASSERT (0);
```

```
;  1129          1728  3                    TES;
;  1130          1729  3
;  1131        L 1730  3 %IF %BLISS (BLISS32)
;  1132          1731  3 %THEN
;  1133          1732  3
;  1134          1733  4                    IF ( NOT .IO_STS)
;  1135          1734  3                    THEN
;  1136          1735  3                        SIGNAL_STOP (SHR$_CLOSEOUT + (EDT$K_FAC_NO*65536) + STS$K_SEVERE, 1,
;  1137          1736  3                            .DESC_ADDR, .IO_STS, .IO_STV);
;  1138          1737  3
;  1139          1738  3                    STR$FREE1_DX (.DESC_ADDR);
;  1140          1739  3 %FI
;  1141          1740  3
;  1142          1741  3                    RETURN (.IO_STS);
;  1143          1742  2                    END;
;  1144          1743  2
;  1145          1744  2                [INRANGE, OUTRANGE] :
;  1146          1745  2                    ASSERT (0);
;  1147          1746  2                TES;
;  1148          1747  2
;  1149          1748  2            ASSERT (0);
;  1150          1749  2            RETURN (0);
;  1151          1750  1            END;                                    ! of routine EDT$FILEIO

                                                .TITLE  EDT$FILEIO FILEIO - Central file I/O module
                                                .IDENT  \V04-000\

                                                .PSECT  _EDT$DATA,NOEXE,  PIC,2

                0000  00000 CMD_DESC:
                                                .WORD   0
                  02  0E  00002                .BYTE   14, 2                   ;
            00000000  00004                     .LONG   0                       ;
                0000  00008 JOU_DESC:
                                                .WORD   0
                  02  0E  0000A                .BYTE   14, 2                   ;
            00000000  0000C                     .LONG   0                       ;
                0000  00010 INP_DESC:
                                                .WORD   0
                  02  0E  00012                .BYTE   14, 2                   ;
            00000000  00014                     .LONG   0                       ;
                0000  00018 ALT_DESC:
                                                .WORD   0
                  02  0E  0001A                .BYTE   14, 2                   ;
            00000000  0001C                     .LONG   0                       ;
                0000  00020 OUT_DESC:
                                                .WORD   0
                  02  0E  00022                .BYTE   14, 2                   ;
            00000000  00024                     .LONG   0                       ;
                      00028 INP_NAME:
                                                .BLKB   256
                      00128 INP_NAME_LEN:
                                                .BLKB   4
                      0012C OUT_IFI:            .BLKB   4
                      00130 JOU_IFI:            .BLKB   4
                      00134 INCL_IFI:
```

```
                                                        .BLKB   4
                                                00138 INP_IFI:.BLKB   4
                                                0013C CMD_IFI:.BLKB   4
                                                00140 DISK_FI:.BLKB   4
                                        00000000  00144 FLUSH_COUNTER:
                                                        .LONG   0                                       ;
                                                00148 INCL_VFC:
                                                        .BLKB   4
                                                0014C INPUT_VFC:
                                                        .BLKB   4

                                                        .PSECT  _EDT$CODE,NOWRT,  SHR,  PIC,2

                                    50  4D  54  2E  00000 P.AAA:  .ASCII  \.TMP\                          ;

                                                        EDT$K_OPEN_INPUT==  1
                                                        EDT$K_OPEN_OUTPUT_SEQ==
                                                                            2
                                                        EDT$K_OPEN_OUTPUT_NOSEQ==
                                                                            3
                                                        EDT$K_OPEN_IN_OUT== 4
                                                        EDT$K_GET==         5
                                                        EDT$K_PUT==         6
                                                        EDT$K_CLOSE_DEL==   7
                                                        EDT$K_CLOSE==       8
                                                        EDT$K_COMMAND_FILE==1
                                                        EDT$K_INPUT_FILE==  2
                                                        EDT$K_INCLUDE_FILE==3
                                                        EDT$K_JOURNAL_FILE==4
                                                        EDT$K_OUTPUT_FILE== 5
                                                        EDT$K_WRITE_FILE==  6
                                                        .EXTRN  EDT$$PAR_FNAME, EDT$$CNV_UPC
                                                        .EXTRN  EDT$$REN_FI, EDT$$FLUSH_OBUF
                                                        .EXTRN  EDT$$OPN_IFIDEF
                                                        .EXTRN  EDT$$OPN_OFIDEF
                                                        .EXTRN  EDT$$CLS_FI, EDT$$RD_IFI
                                                        .EXTRN  EDT$$WR_OFI, STR$FREE1_DX
                                                        .EXTRN  EDT$$OPN_INOUT, STR$COPY_DX
                                                        .EXTRN  STR$COPY_R, EDT$$Z_SYS_PRIRAB
                                                        .EXTRN  EDT$$Z_SYS_JOURAB
                                                        .EXTRN  EDT$$Z_SYS_CMDRAB
                                                        .EXTRN  EDT$$Z_SYS_ALTRAB
                                                        .EXTRN  EDT$_INPFILOPN, EDT$_FILNAM
                                                        .EXTRN  EDT$_INTERERR, EDT$_COMFILNEX
                                                        .EXTRN  EDT$_COMFILNOP, EDT$_NOJNLFIL
                                                        .EXTRN  EDT$_INPFILNEX, EDT$_OUTFILCRE
                                                        .EXTRN  EDT$_NONSTDFIL, EDT$$INTER_ERR

                                                0FFC 00000      .ENTRY  EDT$FILEIO, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 0818
                                                                        R10,R11
                                    5B 00000000G  00  9E 00002      MOVAB   EDT$$Z_SYS_ALTRAB, R11                  ;
                                    5A 00000000G  00  9E 00009      MOVAB   LIB$STOP, R10                           ;
                                    59 00000000'  EF  9E 00010      MOVAB   ALT_DESC, R9                            ;
                                    5E            14  C2 00017      SUBL2   #20, SP                                 ;
                                    56        0C  AC  D0 0001A      MOVL    FILE_REC, R6                            ; 0995
                                    52        10  AC  D0 0001E      MOVL    FILE_RHB, R2                            ; 0996
                                07  01        04  BC  CF 00022      CASEL   @FILECODE, #1, #7                       ; 1002
```

```
     025F        0125        0125        0012      00027 1$:     .WORD    2$-1$,-                                 ;
     03D8        04ED        0346        02B6      0002F                  15$-1$,-                                ;
                                                                          15$-1$,-                                ;
                                                                          28$-1$,-                                ;
                                                                          33$-1$,-                                ;
                                                                          41$-1$,-                                ;
                                                                          68$-1$,-                                ;
                                                                          51$-1$                                  ;
                                                     0E    11 00037       BRB      4$                             ; 1745
                                            08       AE    D4 00039 2$:  CLRL     NONSTD                         ; 1019
                   02            01         08       BC    CF 0003C       CASEL    @FILESTRM, #1, #2              ; 1021
                 00C2          005C        0009            00041 3$:     .WORD    5$-3$,-                        ;
                                                                          8$-3$,-                                 ;
                                                                          12$-3$                                  ;
                                                   0289    31 00047 4$:  BRW      32$                            ; 1140
                                            08       AE    9F 0004A 5$:  PUSHAB   NONSTD                         ; 1029
                                            08       AE    9F 0004D       PUSHAB   VFC                            ;
                                            14       AE    9F 00050       PUSHAB   IO_STV                         ;
                                            1C       AE    9F 00053       PUSHAB   IO_STS                         ;
                                                     7E    D4 00056       CLRL     -(SP)                          ;
                               7E                    62    3C 00058       MOVZWL   (R2), -(SP)                    ; 1030
                                            04       A2    DD 0005B       PUSHL    4(R2)                          ; 1029
                                                     56    DD 0005E       PUSHL    R6                             ;
                                     00000000G       00    9F 00060       PUSHAB   EDT$$Z_SYS_CMDRAB              ;
                      00000000G        00            09    FB 00066       CALLS    #9, EDT$$OPN_IFIDEF            ;
                           0124        C9            50    D0 0006D       MOVL     R0, CMD_IFI                    ;
                                                     13    12 00072       BNEQ     6$                             ; 1035
                                            0C       AE    DD 00074       PUSHL    IO_STV                         ; 1041
                                            14       AE    DD 00077       PUSHL    IO_STS                         ;
                                                     56    DD 0007A       PUSHL    R6                             ; 1040
                                                     01    DD 0007C       PUSHL    #1                             ;
                                     0085109C        8F    DD 0007E       PUSHL    #8720540                       ;
                               6A                    05    FB 00084       CALLS    #5, LIB$STOP                   ;
                               08           08       AE    E9 00087 6$:  BLBC     NONSTD, 7$                     ; 1047
                         10    AE    00000000G       8F    D0 0008B       MOVL     #EDT$_NONSTDFIL, IO_STS        ;
                                            04       A6    DD 00093 7$:  PUSHL    4(R6)                          ; 1052
                                                     56    DD 00096       PUSHL    R6                             ;
                                            E8       A9    9F 00098       PUSHAB   CMD_DESC                       ;
                                                     63    11 0009B       BRB      11$                            ;
                                            08       AE    9F 0009D 8$:  PUSHAB   NONSTD                         ; 1067
                                          0134       C9    9F 000A0       PUSHAB   INPUT_VFC                      ;
                                            14       AE    9F 000A4       PUSHAB   IO_STV                         ;
                                            1C       AE    9F 000A7       PUSHAB   IO_STS                         ;
                                                     7E    D4 000AA       CLRL     -(SP)                          ;
                               7E                    62    3C 000AC       MOVZWL   (R2), -(SP)                    ; 1068
                                            04       A2    DD 000AF       PUSHL    4(R2)                          ; 1067
                                                     56    DD 000B2       PUSHL    R6                             ;
                                     00000000G       00    9F 000B4       PUSHAB   EDT$$Z_SYS_PRIRAB              ;
                      00000000G        00            09    FB 000BA       CALLS    #9, EDT$$OPN_IFIDEF            ;
                           0120        C9            50    D0 000C1       MOVL     R0, INP_IFI                    ;
                           0110        C9            66    3C 000C6       MOVZWL   (R6), INP_NAME_LEN             ; 1072
              10     A9      04        B6    0110    C9    28 000CB       MOVC3    INP_NAME_LEN, @4(R6), INP_NAME ; 1073
                                          0120       C9    D5 000D3       TSTL     INP_IFI                        ; 1078
                                                     13    12 000D7       BNEQ     9$                             ;
                                            0C       AE    DD 000D9       PUSHL    IO_STV                         ; 1081
                                            14       AE    DD 000DC       PUSHL    IO_STS                         ;
                                                     56    DD 000DF       PUSHL    R6                             ; 1080
```

```
                                   01  DD 000E1          PUSHL   #1                              ;
                       0085109C    8F  DD 000E3          PUSHL   #8720540                        ;
                  6A               05  FB 000E9          CALLS   #5, LIB$STOP                    ;
                  08         08    AE  E9 000EC  9$:     BLBC    NONSTD, 10$                     ; 1087
            10    AE  00000000G    8F  D0 000F0          MOVL    #EDT$_NONSTDFIL, IO_STS         ;
                             04    A6  DD 000F8  10$:    PUSHL   4(R6)                           ; 1098
                                   56  DD 000FB          PUSHL   R6                              ;
                             F8    A9  9F 000FD          PUSHAB  INP_DESC                        ;
                                 01C6  31 00100  11$:    BRW     31$                             ;
                             08    AE  9F 00103  12$:    PUSHAB  NONSTD                          ; 1108
                           0130    C9  9F 00106          PUSHAB  INCL_VFC                        ;
                             14    AE  9F 0010A          PUSHAB  IO_STV                          ;
                             1C    AE  9F 0010D          PUSHAB  IO_STS                          ;
                                   01  DD 00110          PUSHL   #1                              ;
                  7E               62  3C 00112          MOVZWL  (R2), -(SP)                     ; 1109
                             04    A2  DD 00115          PUSHL   4(R2)                           ; 1108
                                   56  DD 00118          PUSHL   R6                              ;
                                   5B  DD 0011A          PUSHL   R11                             ;
        00000000G  00              09  FB 0011C          CALLS   #9, EDT$$OPN_IFIDEF             ;
             011C  C9              50  D0 00123          MOVL    R0, INCL_IFI                    ;
                                   13  12 00128          BNEQ    13$                             ; 1111
                             0C    AE  DD 0012A          PUSHL   IO_STV                          ; 1117
                             14    AE  DD 0012D          PUSHL   IO_STS                          ;
                                   56  DD 00130          PUSHL   R6                              ; 1116
                                   01  DD 00132          PUSHL   #1                              ;
                       0085109C    8F  DD 00134          PUSHL   #8720540                        ;
                  6A               05  FB 0013A          CALLS   #5, LIB$STOP                    ;
                  08         08    AE  E9 0013D  13$:    BLBC    NONSTD, 14$                     ; 1123
            10    AE  00000000G    8F  D0 00141          MOVL    #EDT$_NONSTDFIL, IO_STS         ;
                                 00F5  31 00149  14$:    BRW     26$                             ; 1128
                  02         04    BC  D1 0014C  15$:    CMPL    @FILECODE, #2                   ; 1159
                                   05  12 00150          BNEQ    16$                             ;
                  58               01  D0 00152          MOVL    #1, SEQ                         ; 1161
                                   02  11 00155          BRB     17$                             ;
                                   58  D4 00157  16$:    CLRL    SEQ                             ; 1163
      02          04         08    BC  CF 00159  17$:    CASEL   @FILESTRM, #4, #2               ; 1165
    0009        0009             00EC     0015E  18$:    .WORD   27$-18$,-                       ;
                                                                 19$-18$,-                       ;
                                                                 19$-18$                         ;
                                 016C  31 00164          BRW     32$                             ; 1352
                  05         08    BC  D1 00167  19$:    CMPL    @FILESTRM, #5                   ; 1176
                                   0A  12 0016B          BNEQ    20$                             ;
                                   57  D4 0016D          CLRL    ATT                             ; 1179
                  55               01  D0 0016F          MOVL    #1, RELAT                       ; 1180
                  53               01  D0 00172          MOVL    #1, FORCE_MAXV                  ; 1181
                                   07  11 00175          BRB     21$                             ; 1176
                  57               01  D0 00177  20$:    MOVL    #1, ATT                         ; 1185
                                   55  D4 0017A          CLRL    RELAT                           ; 1186
                                   53  D4 0017C          CLRL    FORCE_MAXV                      ; 1187
                                   62  B5 0017E  21$:    TSTW    (R2)                            ; 1193
                                   07  13 00180          BEQL    22$                             ;
        00000000G  00              00  FB 00182          CALLS   #0, EDT$$INTER_ERR              ;
                           0128    C9  D4 00189  22$:    CLRL    DISK_FI                         ; 1194
                  01               55  D1 0018D          CMPL    RELAT, #1                       ; 1204
                                   14  12 00190          BNEQ    23$                             ;
                                   66  B5 00192          TSTW    (R6)                            ;
                                   10  12 00194          BNEQ    23$                             ;
```

```
                                10   A9  9F 00196          PUSHAB  INP_NAME                                ; 1207
                              0110   C9  9F 00199          PUSHAB  INP_NAME_LEN                            ;
                                     56  DD 0019D          PUSHL   R6                                      ;
                00000000G     00     03  FB 0019F          CALLS   #3, STR$COPY_R                          ;
                                  0C   AE  9F 001A6 23$:    PUSHAB  IO_STV                                  ; 1214
                                  14   AE  9F 001A9          PUSHAB  IO_STS                                  ;
                              0128   C9  9F 001AC          PUSHAB  DISK_FI                                 ;
                                     55  DD 001B0          PUSHL   RELAT                                   ;
                                     56  DD 001B2          PUSHL   R6                                      ;
                                     5B  DD 001B4          PUSHL   R11                                     ;
                00000000G     00     06  FB 001B6          CALLS   #6, EDT$$PAR_FNAME                      ;
                                     50  DO 001BD          MOVL    R0, STATUS                              ;
                                     54  E8 001C0          BLBS    STATUS, 24$                             ; 1216
                                  0C   AE  DD 001C3          PUSHL   IO_STV                                  ; 1219
                                  14   AE  DD 001C6          PUSHL   IO_STS                                  ;
                                     56  DD 001C9          PUSHL   R6                                      ; 1218
                                     01  DD 001CB          PUSHL   #1                                      ;
                           008510A4     8F  DD 001CD          PUSHL   #8720548                                ;
                                  6A     05  FB 001D3          CALLS   #5, LIB$STOP                            ;
                                  08   A9  B4 001D6 24$:    CLRW    OUT_DESC                                ; 1221
                                  0C   A9  D4 001D9          CLRL    OUT_DESC+4                              ; 1222
                                  26 0128   C9  E9 001DC          BLBC    DISK_FI, 25$                            ; 1228
                                  04   A6  DD 001E1          PUSHL   4(R6)                                   ; 1231
                                     56  DD 001E4          PUSHL   R6                                      ;
                                  08   A9  9F 001E6          PUSHAB  OUT_DESC                                ;
                00000000G     00     03  FB 001E9          CALLS   #3, STR$COPY_R                          ;
                              FE08   CF  9F 001F0          PUSHAB  P.AAA                                   ; 1232
                        04    AE       04  DO 001F4          MOVL    #4, 4(SP)                               ;
                                  04   AE  9F 001F8          PUSHAB  4(SP)                                   ;
                                     56  DD 001FB          PUSHL   R6                                      ;
                00000000G     00     03  FB 001FD          CALLS   #3, STR$COPY_R                          ;
                                     53  01  DO 00204          MOVL    #1, FORCE_MAXV                          ; 1233
                                  0C   AE  9F 00207 25$:    PUSHAB  IO_STV                                  ; 1240
                                  14   AE  9F 0020A          PUSHAB  IO_STS                                  ;
                                     53  DD 0020D          PUSHL   FORCE_MAXV                              ; 1241
                              00A0   8F  BB 0020F          PUSHR   #^M<R5,R7>                              ;
                                     58  DD 00213          PUSHL   SEQ                                     ;
                                  7E   08  A9  3C 00215          MOVZWL  OUT_DESC, -(SP)                         ;
                                  0C   A9  DD 00219          PUSHL   OUT_DESC+4                              ; 1240
                                     56  DD 0021C          PUSHL   R6                                      ;
                                     5B  DD 0021E          PUSHL   R11                                     ;
                00000000G     00     0A  FB 00220          CALLS   #10, EDT$$OPN_OF_IDEF                   ;
                        0114    C9       50  DO 00227          MOVL    R0, OUT_IFI                             ;
                                     13  12  00022C          BNEQ    26$                                     ; 1246
                                  0C   AE  DD 0022E          PUSHL   IO_STV                                  ; 1249
                                  14   AE  DD 00231          PUSHL   IO_STS                                  ;
                                     56  DD 00234          PUSHL   R6                                      ; 1248
                                     01  DD 00236          PUSHL   #1                                      ;
                           008510A4     8F  DD 00238          PUSHL   #8720548                                ;
                                  6A     05  FB 0023E          CALLS   #5, LIB$STOP                            ;
                                  04   A6  DD 00241 26$:    PUSHL   4(R6)                                   ; 1254
                                     56  DD 00244          PUSHL   R6                                      ;
                                     59  DD 00246          PUSHL   R9                                      ;
                                     7F  11  00248          BRB     31$                                     ;
                                  0C   AE  9F 0024A 27$:    PUSHAB  IO_STV                                  ; 1330
                                  14   AE  9F 0024D          PUSHAB  IO_STS                                  ;
                                     01  DD 00250          PUSHL   #1                                      ;
```

```
                                        02  DD 00252          PUSHL   #2
                                        01  DD 00254          PUSHL   #1
                                        7E  D4 00256          CLRL    -(SP)
                                    7E  62  3C 00258          MOVZWL  (R2), -(SP)                 ;1331
                                    04  A2  DD 0025B          PUSHL   4(R2)                       ;1330
                                        56  DD 0025E          PUSHL   R6
                                00000000G   00  9F 00260      PUSHAB  EDT$$Z_SYS_JOURAB
                          00000000G     00  0A  FB 00266      CALLS   #10, EDT$$OPN_OFIDEF
                              0118      C9      50  D0 0026D  MOVL    R0, JOU_IFI
                                            4D  12 00272      BNEQ    30$                         ;1333
                                        0C  AE  DD 00274      PUSHL   IO_STV                      ;1336
                                        14  AE  DD 00277      PUSHL   IO_STS
                                        56  DD 0027A          PUSHL   R6                          ;1335
                                        01  DD 0027C          PUSHL   #1
                                008510A4    8F  DD 0027E      PUSHL   #8720548
                                            38  11 00284      BRB     29$
                                    04  08  BC  D1 00286 28$:  CMPL    @FILESTRM, #4               ;1366
                                            47  12 0028A      BNEQ    32$
                                        0C  AE  9F 0028C      PUSHAB  IO_STV                      ;1372
                                        14  AE  9F 0028F      PUSHAB  IO_STS
                                    7E  62  3C 00292          MOVZWL  (R2), -(SP)                 ;1373
                                    04  A2  DD 00295          PUSHL   4(R2)                       ;1372
                                        56  DD 00298          PUSHL   R6
                                00000000G   00  9F 0029A      PUSHAB  EDT$$Z_SYS_JOURAB
                          00000000G     00  06  FB 002A0      CALLS   #6, EDT$$OPN_INOUT
                              0118      C9      50  D0 002A7  MOVL    R0, JOU_IFI
                                            13  12 002AC      BNEQ    30$                         ;1375
                                        0C  AE  DD 002AE      PUSHL   IO_STV                      ;1378
                                        14  AE  DD 002B1      PUSHL   IO_STS
                                        56  DD 002B4          PUSHL   R6                          ;1377
                                        01  DD 002B6          PUSHL   #1
                                0085109C    8F  DD 002B8      PUSHL   #8720540
                                    6A  05  FB 002BE 29$:      CALLS   #5, LIB$STOP
                                    04  A6  DD 002C1 30$:      PUSHL   4(R6)                       ;1380
                                        56  DD 002C4          PUSHL   R6
                                        F0  A9  9F 002C6      PUSHAB  JOU_DESC
                          00000000G     00  03  FB 002C9 31$:  CALLS   #3, STR$COPY_R
                                          02AA  31 002D0      BRW     76$                         ;1387
                          00000000G     00      00  FB 002D3 32$:  CALLS   #0, EDT$$INTER_ERR      ;1390
                                          02A5  31 002DA      BRW     77$                         ;1002
                              03          01  08  BC  CF 002DD 33$:  CASEL   @FILESTRM, #1, #3     ;1401
                0042        0034        0021        0011      002E2 34$:  .WORD   35$-34$,-
                                                                            36$-34$,-
                                                                            37$-34$,-
                                                                            38$-34$
                          00000000G     00      00  FB 002EA      CALLS   #0, EDT$$INTER_ERR      ;1433
                                            3F  11 002F1      BRB     39$                         ;1401
                                    53      E8  A9  9E 002F3 35$:  MOVAB   CMD_DESC, DESC_ADDR     ;1406
                                    50 00000000G    00  9E 002F7  MOVAB   EDT$$Z_SYS_CMDRAB, RAB  ;1407
                                        04  AE  D4 002FE      CLRL    VFC                         ;1408
                                            2F  11 00301      BRB     39$                         ;1401
                                    53      F8  A9  9E 00303 36$:  MOVAB   INP_DESC, DESC_ADDR     ;1413
                                04  AE      0134    C9  D0 00307  MOVL    INPOT_VFC, VFC          ;1414
                                    50 00000000G    00  9E 0030D  MOVAB   EDT$$Z_SYS_PRIRAB, RAB  ;1415
                                            1C  11 00314      BRB     39$                         ;1401
                                04  AE      0130    C9  D0 00316 37$:  MOVL    INCL_VFC, VFC       ;1420
                                    53          69  9E 0031C  MOVAB   ALT_DESC, DESC_ADDR         ;1421
```

```
                                50              6B  9E 0031F               MOVAB   EDT$$Z_SYS_ALTRAB, RAB                  ; 1422
                                                0E  11 00322               BRB     39$                                     ; 1401
                                        04      AE  D4 00324 38$:          CLRL    VFC                                     ; 1427
                                53      F0      A9  9E 00327               MOVAB   JOU_DESC, DESC_ADDR                     ; 1428
                                50 00000000G    00  9E 0032B               MOVAB   EDT$$Z_SYS_JOURAB, RAB                  ; 1429
                                        04      AE  DD 00332 39$:          PUSHL   VFC                                     ; 1438
                                        10      AE  9F 00335               PUSHAB  IO_STV
                                        18      AE  9F 00338               PUSHAB  IO_STS
                                                52  DD 0033B               PUSHL   R2
                                      0041      8F  BB 0033D               PUSHR   #^M<R0,R6>
                    00000000G   00              06  FB 00341               CALLS   #6, EDT$$RD_IFI
                                54              50  D0 00348               MOVL    R0, STATUS
                                72              54  E8 0034B               BLBS    STATUS, 47$                             ; 1440
                    0001827A    8F          10  AE  D1 0034E               CMPL    IO_STS, #98938                          ; 1443
                                                03  12 00356               BNEQ    40$
                                              0222  31 00358               BRW     76$
                                        0C      AE  DD 0035B 40$:          PUSHL   IO_STV                                  ; 1448
                                        14      AE  DD 0035E               PUSHL   IO_STS                                  ; 1447
                                                53  DD 00361               PUSHL   DESC_ADDR
                                                01  DD 00363               PUSHL   #1
                                  008510B4      8F  DD 00365               PUSHL   #8720564
                                                50  11 0036B               BRB     46$
                  02                04          08  BC  CF 0036D 41$:      CASEL   @FILESTRM, #4, #2                       ; 1473
                0000F             0000F               0017    00372 42$:     .WORD   44$-42$,-
                                                                                   43$-42$,-
                                                                                   43$-42$
                    00000000G   00              00  FB 00378               CALLS   #0, EDT$$INTER_ERR                      ; 1489
                                                13  11 0037F               BRB     45$                                     ; 1473
                                55              69  9E 00381 43$:          MOVAB   ALT_DESC, DESC_ADDR                     ; 1478
                                53              6B  9E 00384               MOVAB   EDT$$Z_SYS_ALTRAB, RAB                  ; 1479
                                                0B  11 00387               BRB     45$                                     ; 1473
                                55      F0      A9  9E 00389 44$:          MOVAB   JOU_DESC, DESC_ADDR                     ; 1484
                                53 00000000G    00  9E 0038D               MOVAB   EDT$$Z_SYS_JOURAB, RAB                  ; 1485
                                        0C      AE  9F 00394 45$:          PUSHAB  IO_STV                                  ; 1494
                                        14      AE  9F 00397               PUSHAB  IO_STS
                                                52  DD 0039A               PUSHL   R2
                                      0048      8F  BB 0039C               PUSHR   #^M<R3,R6>
                00000000G       00              05  FB 003A0               CALLS   #5, EDT$$WR_OFI
                                54              50  D0 003A7               MOVL    R0, STATUS
                                15              54  E8 003AA               BLBS    STATUS, 48$                             ; 1500
                                        0C      AE  DD 003AD               PUSHL   IO_STV                                  ; 1506
                                        14      AE  DD 003B0               PUSHL   IO_STS                                  ; 1505
                                                55  DD 003B3               PUSHL   DESC_ADDR
                                                01  DD 003B5               PUSHL   #1
                                  008510D4      8F  DD 003B7               PUSHL   #8720596
                                6A              05  FB 003BD 46$:          CALLS   #5, LIB$STOP
                                                39  11 003C0 47$:          BRB     50$
                                04      08      BC  D1 003C2 48$:          CMPL    @FILESTRM, #4                           ; 1511
                                                33  12 003C6               BNEQ    50$
                                      012C      C9  D6 003C8               INCL    FLUSH_COUNTER                           ; 1514
                                05    012C      C9  D1 003CC               CMPL    FLUSH_COUNTER, #5                       ; 1516
                                                28  12 003D1               BNEQ    50$
                                        0C      AE  9F 003D3               PUSHAB  IO_STV                                  ; 1522
                                                53  DD 003D6               PUSHL   RAB
                00000000G       00              02  FB 003D8               CALLS   #2, EDT$$FLUSH_OBUF
                                54              50  D0 003DF               MOVL    R0, STATUS
                                12              54  E8 003E2               BLBS    STATUS, 49$                             ; 1524
```

```
                                  0C    AE  DD 003E5         PUSHL   IO_STV                    ; 1527
                                        54  DD 003E8         PUSHL   STATUS                    ;
                                        55  DD 003EA         PUSHL   DESC_ADDR                 ; 1526
                                        01  DD 003EC         PUSHL   #1                        ;
                          008510D4      8F  DD 003EE         PUSHL   #8720596                  ;
                               6A       05  FB 003F4         CALLS   #5, LIB$STOP              ;
                                  012C  C9  D4 003F7  49$:   CLRL    FLUSH_COUNTER             ; 1533
                               50       54  D0 003FB  50$:   MOVL    STATUS, R0                ; 1538
                                            04 003FE         RET                               ;
                    05         01   08  BC  CF 003FF  51$:   CASEL   @FILESTRM, #1, #5         ; 1548
       00D6       0054       0034     0015     00404  52$:   .WORD   53$-52$,-                 ;
                             0075     0075     0040C                 54$-52$,-                 ;
                                                                     56$-52$,-                 ;
                                                                     64$-52$,-                 ;
                                                                     58$-52$,-                 ;
                                                                     58$-52$                   ;
                 00000000G  00          00  FB 00410         CALLS   #0, EDT$$INTER_ERR        ; 1671
                                        5E  11 00417         BRB     57$                       ; 1548
                            52    E8    A9  9E 00419  53$:   MOVAB   CMD_DESC, DESC_ADDR       ; 1556
                            53  1050    8F  3C 0041D         MOVZWL  #4176, ERROR              ; 1557
                                  0C    AE  9F 00422         PUSHAB  IO_STV                    ; 1558
                                  14    AE  9F 00425         PUSHAB  IO_STS                    ;
                                        52  DD 00428         PUSHL   DESC_ADDR                 ;
                                        7E  D4 0042A         CLRL    -(SP)                     ;
                          00000000G     00  9F 0042C         PUSHAB  EDT$$Z_SYS_CMDRAB         ;
                                  0124  C9  DD 00432         PUSHL   CMD_IFI                   ;
                                        1D  11 00436         BRB     55$                       ;
                            52    F8    A9  9E 00438  54$:   MOVAB   INP_DESC, DESC_ADDR       ; 1570
                            53  1050    8F  3C 0043C         MOVZWL  #4176, ERROR              ; 1571
                                  0C    AE  9F 00441         PUSHAB  IO_STV                    ; 1572
                                  14    AE  9F 00444         PUSHAB  IO_STS                    ;
                                        52  DD 00447         PUSHL   DESC_ADDR                 ;
                                        7E  D4 00449         CLRL    -(SP)                     ;
                          00000000G     00  9F 0044B         PUSHAB  EDT$$Z_SYS_PRIRAB         ;
                                  0120  C9  DD 00451         PUSHL   INP_IFI                   ;
                                    009F    31 00455  55$:   BRW     65$                       ;
                            52          69  9E 00458  56$:   MOVAB   ALT_DESC, DESC_ADDR       ; 1584
                                  0C    AE  9F 0045B         PUSHAB  IO_STV                    ; 1585
                                  14    AE  9F 0045E         PUSHAB  IO_STS                    ;
                                        52  DD 00461         PUSHL   DESC_ADDR                 ;
                                        7E  D4 00463         CLRL    -(SP)                     ;
                                        5B  DD 00465         PUSHL   R11                       ;
                                  011C  C9  DD 00467         PUSHL   INCL_IFI                  ;
                 00000000G  00          06  FB 0046B         CALLS   #6, EDT$$CLS_FI           ;
                            53  1050    8F  3C 00472         MOVZWL  #4176, ERROR              ; 1586
                                        5F  11 00477  57$:   BRB     63$                       ; 1548
                            05    08    BC  D1 00479  58$:   CMPL    @FILESTRM, #5             ; 1599
                                        05  12 0047D         BNEQ    59$                       ;
                            54          01  D0 0047F         MOVL    #1, FORCE_MAXV            ;
                                        02  11 00482         BRB     60$                       ;
                                        54  D4 00484  59$:   CLRL    FORCE_MAXV                ;
                            52          69  9E 00486  60$:   MOVAB   ALT_DESC, DESC_ADDR       ; 1603
                            53  1058    8F  3C 00489         MOVZWL  #4184, ERROR              ; 1604
                                  0C    AE  9F 0048E         PUSHAB  IO_STV                    ; 1605
                                  14    AE  9F 00491         PUSHAB  IO_STS                    ;
                                        52  DD 00494         PUSHL   DESC_ADDR                 ;
                                        7E  D4 00496         CLRL    -(SP)                     ;
```

```
                                5B  DD 00498          PUSHL   R11                               ;
                        0114    C9  DD 0049A          PUSHL   OUT_IFI                           ;
         00000000G  00          06  FB 0049E          CALLS   #6, EDT$$CLS_FI                   ;
                    59    10    AE  E9 004A5          BLBC    IO_STS, 67$                       ; 1610
                    1C  0128    C9  E9 004A9          BLBC    DISK_FI, 61$                      ; 1613
                          0C    AE  9F 004AE          PUSHAB  IO_STV                            ; 1616
                          14    AE  9F 004B1          PUSHAB  IO_STS                            ;
                                54  DD 004B4          PUSHL   FORCE_MAXV                        ;
                          08    A9  9F 004B6          PUSHAB  OUT_DESC                          ;
                                59  DD 004B9          PUSHL   R9                                ;
         00000000G  00          05  FB 004BB          CALLS   #5, EDT$$REN_FI                   ;
                          0C    A9  DD 004C2          PUSHL   OUT_DESC+4                        ; 1617
                          08    A9  9F 004C5          PUSHAB  OUT_DESC                          ;
                                05  11 004C8          BRB     62$                               ;
                          04    A9  DD 004CA 61$:     PUSHL   ALT_DESC+4                        ; 1620
                                59  DD 004CD          PUSHL   R9                                ;
                                56  DD 004CF 62$:     PUSHL   R6                                ;
         00000000G  00          03  FB 004D1          CALLS   #3, STR$COPY_R                    ;
                                24  11 004D8 63$:     BRB     66$                               ; 1613
                    52    F0    A9  9E 004DA 64$:     MOVAB   JOU_DESC, DESC_ADDR               ; 1661
                    53  1058    8F  3C 004DE          MOVZWL  #4184, ERROR                      ; 1662
                          0C    AE  9F 004E3          PUSHAB  IO_STV                            ; 1663
                          14    AE  9F 004E6          PUSHAB  IO_STS                            ;
                                52  DD 004E9          PUSHL   DESC_ADDR                         ;
                                7E  D4 004EB          CLRL    -(SP)                             ;
                   00000000G    00  9F 004ED          PUSHAB  EDT$$Z_SYS_JOURAB                 ;
                        0118    C9  DD 004F3          PUSHL   JOU_IFI                           ;
         00000000G  00          06  FB 004F7 65$:     CALLS   #6, EDT$$CLS_FI                   ;
                    72    10    AE  E8 004FE 66$:     BLBS    IO_STS, 75$                       ; 1680
                          0C    AE  DD 00502 67$:     PUSHL   IO_STV                            ; 1683
                          14    AE  DD 00505          PUSHL   IO_STS                            ;
                                52  DD 00508          PUSHL   DESC_ADDR                         ; 1682
                                01  DD 0050A          PUSHL   #1                                ;
                    00850004    E3  9F 0050C          PUSHAB  8716292(ERROR)                    ;
                                5D  11 00512          BRB     74$                               ;
           02       04    08    BC  CF 00514 68$:     CASEL   @FILESTRM, #4, #2                 ; 1697
         000F     000F      0024       00519 69$:     .WORD   71$-69$,-                         ;
                                                              70$-69$,-                         ;
                                                              70$-69$                           ;
         00000000G  00          00  FB 0051F          CALLS   #0, EDT$$INTER_ERR                ; 1727
                                35  11 00526          BRB     73$                               ; 1697
                    52          69  9E 00528 70$:     MOVAB   ALT_DESC, DESC_ADDR               ; 1705
                          0C    AE  9F 0052B          PUSHAB  IO_STV                            ; 1706
                          14    AE  9F 0052E          PUSHAB  IO_STS                            ;
                                59  DD 00531          PUSHL   R9                                ;
                                01  DD 00533          PUSHL   #1                                ;
                                5B  DD 00535          PUSHL   R11                               ;
                        0114    C9  DD 00537          PUSHL   OUT_IFI                           ;
                                19  11 0053B          BRB     72$                               ;
                    52    F0    A9  9E 0053D 71$:     MOVAB   JOU_DESC, DESC_ADDR               ; 1718
                          0C    AE  9F 00541          PUSHAB  IO_STV                            ; 1719
                          14    AE  9F 00544          PUSHAB  IO_STS                            ;
                          F0    A9  9F 00547          PUSHAB  JOU_DESC                          ;
                                02  DD 0054A          PUSHL   #2                                ;
                   00000000G    00  9F 0054C          PUSHAB  EDT$$Z_SYS_JOURAB                 ;
                        0118    C9  DD 00552          PUSHL   JOU_IFI                           ;
         00000000G  00          06  FB 00556 72$:     CALLS   #6, EDT$$CLS_FI                   ;
```

```
                                 13        10   AE  E8 0055D 73$:    BLBS    IO_STS, 75$                                    ; 1733
                                           0C   AE  DD 00561         PUSHL   IO_STV                                         ; 1736
                                           14   AE  DD 00564         PUSHL   IO_STS                                         ;
                                           52   DD  00567           PUSHL   DESC_ADDR                                      ;
                                           01   DD  00569           PUSHL   #1                                             ; 1735
                                    0085105C    8F  DD 0056B         PUSHL   #8720476                                       ;
                                    6A          05  FB 00571 74$:    CALLS   #5, LIB$STOP                                   ;
                                           52   DD  00574 75$:      PUSHL   DESC_ADDR                                      ; 1738
                          00000000G 00         01  FB 00576         CALLS   #1, STR$FREE1_DX                               ;
                                    50   10   AE  D0 0057D 76$:      MOVL    IO_STS, R0                                     ; 1741
                                           04  00581               RET                                                    ;
                          00000000G 00         00  FB 00582 77$:    CALLS   #0, EDT$$INTER_ERR                             ; 1748
                                           50  D4  00589           CLRL    R0                                             ; 1749
                                           04  0058B               RET                                                    ; 1750

; Routine Size:  1420 bytes,    Routine Base:  _EDT$CODE + 0004

; 1152          1751  1
; 1153          1752  1 !<BLF/PAGE>
```

```
; 1155        1753  1 END                                   ! of module EDT$FILEIO
; 1156        1754  1
; 1157        1755  0 ELUDOM


                                                            .EXTRN  LIB$STOP

;                               PSECT SUMMARY
;
;       Name                    Bytes                       Attributes
;
;   _EDT$DATA                     336  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;   _EDT$CODE                    1424  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;   .  ABS  .                       0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)


;                           Library Statistics
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;   _$255$DUA28:[EDT.SRC]EDT.L32;1         377         3         0          40       00:00.2
;   _$255$DUA28:[EDT.SRC]PSECTS.L32;1        2         1        50           7       00:00.1
;   _$255$DUA28:[SYSLIB]STARLET.L32;1     9776        14         0         581       00:04.1



;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS$:FILEIO/OBJ=OBJ$:FILEIO MSRC$:FILEIO.BLI/UPDATE=(ENH$:FILEIO)

; Size:          1420 code + 340 data bytes
; Run Time:        01:06.6
; Elapsed Time:    01:23.8
; Lines/CPU Min:    1581
; Lexemes/CPU-Min:  7978
; Memory Used:  357 pages
; Compilation Complete
```

EXTEND
LIS

FDEC
LIS

FILL
LIS

FINDPARA
LIS

FORLF
LIS

EDT
LIS

EXEC
LIS

EXECNOO
LIS

FILEIO
LIS

EDTVECTOR
LIS

FINDKEY
LIS

FINAL
LIS

FINDHNDLR
LIS

DEFKEY
LIS

ERRMSG
LIS